UNIVERSITY OF GONDAR

FACULTY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

UNSUPERVISED MACHINE LEARNING APPROACH FOR TIGRIGNA WORD SENSE DISAMBIGUATION

By: Meresa Mebrahtu Reda

Advisor: Solomon Teferra (PhD)

A Thesis Submitted to the faculty of Informatics University of Gondar in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science

March, 2017

Gondar, Ethiopia

UNIVERSITY OF GONDAR

FACULTY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

UNSUPERVISED MACHINE LEARNING APPROACH FOR TIGRIGNA WORD SENSE DISAMBIGUATION

By: Meresa Mebrahtu Reda

Advisor: Solomon Teferra (PhD)

Approved by:

Examining Board:

1. **Solomon Teferra (PhD)**, Advisor _____ _____
2. **External Examiner** _____ _____
3. **Internal Examiner** _____ _____
4. _____ _____ _____

DEDICATION

This research work is dedicated to my brother Girmay Mebrahtu Reda.

Acknowledgements

First and foremost I praise the Almighty God, and St. Marry for giving me health and patience in completing my thesis work.

Next, I would like to express my sincere gratitude to my advisor Dr. Solomon Teferra for the continuous support of this research.

Besides my advisor, I would like to thank my family specially my little sis, Werqie Kelali Roso, for making extremely supportive and encouraging in difficult times to become my life prettier, and also, for passing all those hardships I had to go through easier. I couldn't have been able to make it without her.

I am also highly thankful to my friends Mr. Tsegay G/Meskel, lecturer in the department of computer science in Axum University and, Mr. Mesele Niguse lecturer in the department of Information Technology in Assosa University, who helped me in debugging my documentation and critical bugs while writing the code of the Tigrigna stemmer with Java and Weka.

List of Acronyms

AI	Artificial Intelligence
AL	Average Link
ANNs	Artificial Neural Networks
ARFF	Attribute-Relation File Format
CL	Complete Link
CV	Cross Validation
CV	Consonant Vowel
EM	Expectation Maximization
GHSOM	Growing Hierarchical Self-Organizing Map
IA	Inter-Annotation Agreement
IE	Information Extraction
IR	Information Retrieval
MRD	Machine Readable Dictionary
MT	Machine Translation
NLP	Natural Language Processing
NLU	Natural Language Understanding
OALD	Oxford Advanced Learner's Dictionary
SERA	System for Ethiopic Representation in ASCII
SL	98Single Link
SOV	Subject-Object-Verb
SVM	Support Vector Machine
WEKA	Waikato Environment for Knowledge Analysis
WSD	Word Sense Disambiguation

Abstract

All human languages have words that can mean different things in different contexts. Word sense disambiguation (WSD) is an open problem of natural language processing, which governs the process of identifying which sense of a word (i.e. meaning) is used in a sentence, when the word has multiple meanings (polysemy).

In this paper, we are concerned with a corpus based approach to word sense disambiguation for Tigrigna texts that only requires information that can be automatically extracted from untagged text. We use unsupervised techniques to address the problem of automatically deciding the correct sense of an ambiguous word based on its surrounding context. And we report experiments on four selected Tigrigna ambiguous words due to lack of sufficient training data; these are መደብ read as "medeb" has three different meaning (Program, Traditional bed and Grouping), ሓለፈ read as "halefe"; has four dissimilar meanings (Pass, Promote, Boss and Pass away), ሃደመ read as "hademe"; has two different meaning (Running and Building house) and, ከበረ read as "kebere"; has two different meaning (Respecting and Expensive).

For the purposes of this research, unsupervised machine learning technique was applied to a corpus of Tigrigna sentences so as to acquire disambiguation information automatically. A total of 631 sense examples transcribed to Latin script for the four ambiguous words were collected from different online Tigrigna websites and newspapers.

Finally we tested five clustering algorithms (simple k means, hierarchical agglomerative: Single, Average and complete link and Expectation Maximization algorithms) in the existing implementation of Weka 3.8.1 package. "Use training set" evaluation mode was selected to learn the selected algorithms in the preprocessed dataset. We have evaluated the algorithms for the four ambiguous words and achieved the best accuracy with in the range of 52 to 77.5% for Simple k-means, 67 to 83.3 for EM, 45.6 to 74.1 for Single, 65 to 73.3 for AL and 65 to 73.3 for CL clustering algorithms which is encouraging result.

Finally we achieve the best accuracy 67 to 83.3 in EM algorithm. However, we face challenges in collecting datasets, properly stemming of words and transliterating the sentences to SERA system in order to get higher accuracy. Owing that, further experiments for other ambiguous words and using different approaches needed to better natural language understanding of Tigrigna language.

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER ONE. INTRODUCTION

1.1 Background

There is a need for people all over the World to be able to use their own language when using computers or accessing information on the Internet. This requires the existence of avariety of applications including local language spell-checkers, word processors, machine translation systems, word sense disambiguation, search engines, etc [1]. Tigrigna is a language spoken in the east African countries of Eritrea and Ethiopia. It is one of the two official languages of the State of Eritrea. It is also a working language of the Tigray region of Ethiopia. It is estimated to be spoken by over ten million (see Appendix I) Tigrigna language speakers throughout the world. Tigrigna is a Semitic language of the Afro-asiatic family originated from the ancient Geez language. It is closely related to Amharic and Tigre [29].

Tigrigna language has also a lot of its own words that have more than one lexical meaning or sense; however, usually only one of them is active in a given context [6]. Since the appearance of the first computers, in the earlier 50's, humans have been thinking in Natural Language Understanding (NLU). Since then, lots of talking computers appeared in fiction novels and films. Humans usually see computers as intelligent devices. However,the pioneers of Natural Language Processing (NLP) underestimated the complexity of the task [3].

Word Sense Disambiguation or discourse where this meaning is distinguishable from other senses potentially attributable to that word [28]. WSD is a natural classification problem: given a word and its possible senses, as defined by a dictionary, classify an occurrence of the word in context into two or more of its sense classes. Developing algorithms to replicate this human ability can often be a difficult task [4]. A computer program has no basis for knowing which one is appropriate, even if it is obvious to a human [2].

Nowadays, the advancement of information technology has given birth to the internet that results huge collection of information to address the information need of the society. Although the advantage of the technology keeps up, natural language ambiguities become challenging problems due to scarcity of natural language processing systems in many languages [30]. Tigrigna has been one of the under-resourced languages both in terms of electronic resources and natural language processing tools to access favourable conditions that information technology has brought [29].

WSD is an awkward problem [18]; most problems arise from the fact that the concept of a meaning is vague. Usually, there are no clear boundaries between one sense and the other. Typically, the problem of defining meaning is begun with using dictionaries, which is sense inventory in a context of WSD, i.e., from the algorithmic point of view sense inventories are used to specify all the meanings that a given word has. Now, the goal of WSD can be stated as choosing correct sense from sense inventory in a given context of a word [19].

To address the mentioned challenges and create background solution for natural language processing systems for the language, word sense disambiguation is one of the techniques proposed to reduce or avoid word ambiguities. The application of WSD has great utility of fields including Information Retrieval, Information Extraction and Machine Translation [2,16].

Several approaches have been proposed such as knowledge based approach, suppervised approach, and unsupervised approach for assigning the correct sense to a word in context, some of them achieving remarkable high accuracy figures. Initially, these methods were usually tested only on a small set of words with few and clear sense distinctions [5].

As discussed in [2] in corpus based approaches, information is gained from training on some corpus. A corpus provides a set of samples that enables the systems to develop some numerical models.In supervised WSD the training data is sense-tagged where as in unsupervised WSD the training data is a raw corpora which are not semantically disambiguated. The aim in supervised disambiguation is to build a classifier which correctly classifies new cases based on their context of use .

Current word sense disambiguation (WSD) systems are based on supervised learning methods which is still limited in that it does not work well for all words in a language. One of the main reasons is the lack of sufficient labelled training data that require expertise. Even though one can always label more examples to achieve better performance on a particular data set but the expense can be uncomforted [21]. A major problem with supervised approaches is the need for a large sense-tagged training set .

Unsupervised learning is the greatest challenge for WSD researchers. Unsupervised WSD approaches are composed of word sense induction or discrimination techniques aimed at discovering senses automatically based on unlabeled corpora and then applying them for WSD [5]. Unsupervised methods correspond to clustering tasks rather than sense tagging tasks.

In general, accuracy of unsupervised WSD systems are 5% to 10% lower than that of other algorithms since no lexical resources for training or defining senses are used [15].

Knowledge-based WSD algorithms are similar to Minimally-supervised WSD approaches. The objective of Knowledge-based methods is to exploit static knowledge resources, such as dictionaries, thesauri, glossaries, ontologies, collocation etc., to infer the senses of words in context [5,7,10]. Knowledge-based methods mainly try to avoid the need of large amounts of training materials required in supervised methods [2].

Although different methods have been tested to find the correct sense of the polysemy words, accuracy at satisfactory level has not been obtained yet [15]. Among the different methods used for WSD; this research was focuse its study on exploring unsupervised machine learning approach to WSD for Tigrigna words. Test the results in order to improve a bit further natural language understanding for Tigrigna word disambiguation. Besids, relate the results with other languages has been done before with supervised, and semi_supervised approachs. Unlike that of supervised, unsupervised WSD system deals with grouping of contexts for given word that express the same meaning without providing explicit sense labels for each group (e.g., without using a dictionary) [16].

1.2 Statement of the Problem

Word sense disambiguation is a significant problem at the lexical level of natural language processing. The philosophy is to determine the meaning of a word in a particular usage, by using sense similarity and syntactic context with corpus evidence as well as semantic relations from WordNet [17]. As stated in [4], resolving the ambiguity of words is a central problem for large scale language understanding applications and their associate tasks. Humans are so skilled at resolving potential ambiguities that they do not realize they are doing it. There is considerable focus on how people resolve ambiguities; however it is still not known how exactly humans do lexical disambiguation [15]. Therefore, it is a difficult task to teach a computer to do the same thing. If there are more than one ambiguous words in a sentence, the number of potential interpretations of the sentence increases dramatically [2].

Acording Yarowsky[21], word sense ambiguity is the fundamental problem for many established Human Language Technology applications (e.g., Machine Translation, Information Extraction, Question Answering, Information Retrieval, Text Classification, and Text Summarization). This

is also the reason for associated subtasks (e.g., reference resolution, acquisition of sub categorization patterns, parsing, and, obviously, semantic interpretation). Due to that, many international research groups are working on WSD, using a wide range of approaches. However, current state-of-the-art accuracy is in the range 60–70%, WSD is one of the most important open problems in NLP[21] .

Correctly disambiguating words is a difficult problem [31]. When restricted to available on-line dictionaries like WordNet, it is sometimes impossible even for human beings to pick the right sense for words. Expecting a machine to resolve such ambiguities is not reasonable. But, a good online dictionary with example uses of words in each of their possible senses can allow a machine to disambiguate words accurately. Such dictionaries are not yet available. Incorrect disambiguation not only excludes correct synonyms from the query but it also introduces incorrect information to it reducing retrieval performance [27].

To date, a lot of research works on WSD have been done in English and many other languages such as French, Spanish, Japanese, Hebrew, Chinese and German to facilitate many NLP processes (e.g. Machine Translation (MT), Information Retrival (IR) and Information Extraction(IE), question and answering). Empirical results of these studies indicate that the performance of the systems considerably increased after WSD is applied to them [26]. Solomon [17] was the first researcher that employ supervised machine learning approach for Amharic WSD, and the achieved accuracy was 70% to 83% in training and test set.

Ambiguities have been an issue in researches conducted in Tigrigna language.As discussed earlier; there are many uses for word sense disambiguation. The most common are application of WSD in machine translation, Information retrieval, speech processing, text processing, grammatical analysis, content and thematic analysis. The absence of automatic WSD would make it the development of such NLP and IR applications difficult [17]. A variety of WSD methods have been proposed over the last decade; however, such methods are still immature or undeveloped [25]. In response to this situation, the major concern of this research was to explore unsupervised machine learning approach for WSD to Tigrigna words, examine the outcomes in order to improve a bit further NLU.

For this purpose, this study attempts to address the following research questions:

1) Can unsupervised algorithms improve the performances of Tigrigna WSD using different Clustering algorithms?
2) Exploring the impact of stemming on the effectiveness of unsupervised machine learning algorithms?

1.3 Objective of the study

1.3.1 General Objective

The general objective of this research is to applay unsupervised machine learning techniques for word sense disambiguation to Tigrigna texts.

1.3.2 Specific Objectives

1) To review related literatures available algorithms and techniques of Word Sense Disambiguation
2) To study the possible application of the algorithms and techniques of the Machine Learning field in order to handle the Word Sense Disambiguation task in Tigrigna language
3) To collect and prepare appropriate data sets(corpora) from different sources for training and testing purpose
4) To build WSD model using the selected unsupervised machine learning algorithms
5) To test and evaluate the performance of the prototype issue Word Sense Disambiguation

1.4 Scope and Limitation of the study

In Tigrigna language, there are different sources of disambiguation including lexical, semantic, phonological, referential, syntactic, orthographical etc. The purpose of this research is limited to lexical ambiguity which is concerned only with meanings of individual words. The task attempts to evaluate only four Tigrigna words which have at least two meanings each. Owing unavailability of sense annotated data and linguistic resources such as word net, thearsurs, dictioners and bag of words; the study is limited to the lexical ambiguity of four Tigrigna ambiguous words using unsupervised machine learning algorithm to build and evaluate. Because unsupervised machine learning reduces the bottleneck of labaled data need.

1.5 Significance of the study

As discussed in [26], word sense disambiguation is an "intermediate" task that is necessary for achieving most natural language processing tasks, especially MT and IR. For example for MT, WSD is vital for selecting the suitable target language word for an ambiguous source language word. For IR, it reduces the retrieval of irrelevant documents that contain query words of different senses. In question answering systems, it is used to retrieve the appropriate answer from the document collection for a given query containing ambiguous words. Despite the increasing importance of IR systems as data retrieval tools, the performance of most of these systems has not yet reached a satisfactory level. Word sense ambiguity is one of the reasons for their poor performance.

By improving the accuracy of WSD, can improve performances of Tigrigna language processing tasks including MT, IE, IR, Part of Speech tagging. It also contributes to future researches and development in the area of NLP.

1.6 Methodology of the study

1.6.1 Research Design

An Extensive reviews of literatures related to WSD was conducted in order to investigate the fundamental principles of various approaches, algorithms and tools that would be best for this research and the structure of the documents to be summarized for testing was investigated from review of related literatures. Such as:-

1) Word sense disambiguation
2) Different Machine learning approaches, techniques
3) Ambiguities in Tigrigna language, Tigrigna writing system, punctuation marks and syntactic structure;
4) Different clustering algorithms and their application in machine learning technique

1.6.2 Data Source and Corpus Preparation

WSD systems need well organized data sets for training to make their accuracy attractive [3]. The proposed WSD prototype used corpus to extract a lot of relevant words from it for disambiguation purpose.

For this research, the Tigrigna corpus was collected from resources that can be found on the internet including Tigrigna newspapers and sites (such as: www.wuraynamagazine.com, www.dmtsiweyane.com, www.woyengazeta.org , http://tigrigna.voanews.com/a/ and Tigrigna Bible). The further data enable the researcher to build a larger corpus size that help to come up with a better model.

The collected data was passing through some preprocessing tasks including tokenization, stop word removal and stemming.Using program codes written using java programming language netbeans-8.0.1-windows (jdk-8u66-windows-i586). The preprocessed data is transliterated in to SERA system (System for Ethiopic Representation in ASCII) which maps each Tigrigna Fidel to Latin characters.

Among the list of Ambiguous words, the most widely used and having the same word class of senses are selected. "*Because ambiguous words that have senses with different word classes can be resolved using part of speech tagger by their word class* [17] ". Because of this, this study was prepared containing four (4) ambiguous words to develop the proposed prototype model.

1.7 Tools and Techniques

As most NLP systems, a preliminary preprocessing of the input text is needed. Texts (sentences) preprocessing is a primary step to load the instances of data set into machine learning tool (WEKA) to develop WSD model for the study. The preprocessing task comprises tokenization, stop word removal, stemming and normalization. The preprocessing part of the WSD prototype will accomplish using Java NetBeans-8.0.1-windows; jdk-8u66-windows-i586 software.

In building the WSD model, the researcher was use five unsupervised algorithms that are found in the existing implementation Weka 3.8.1 package. But the researcher trying to choose algorithms representing a few different approaches to the problem of clustering [16]. The researcher was start with simple k-means algorithms, which represent simple, hard and flat clustering methods. The researcher was use agglomerative single, average and complete link algorithms for representative family of hierarchical clustering algorithms. Last but not least, we were test also the Expectation Maximization algorithms also known as the EM which is probabilistic clustering algorithms.

WEKA 3.8.1 machine learning tool is selected due to the familiarity of the researcher to the tool and because of its accessibility, its processing capability and language independent features. Moreover WEKA is available for free on the internet.

Java programming language used to develop the stemmer algorithm for Tigrigna; because it is enormously popular and its rapid rise and wide acceptance can be traced to its design characteristics, particularly its promise that you can write a program once and run it anywhere. Java is a full-featured, general-purpose programming language that can be used to develop robust mission-critical applications. Today, it is employed not only for Web programming, but also for developing standalone applications across platforms on servers, desktop computers, and mobile devices [29]. And Eclipse was used for compiling, running and editing the java program.

1.7.1 Training and Testing

The system was trained 631 sentences and evaluates by "using training set"for the four ambiguous words to create a model. A total of five experiments were carried out using "training set" evaluation technique with different features and its parameters to train the model. Finally, the performances of the clustering algorithm were evaluated using the maximum accuracy of their result.

1.8 Evaluation Techniques

We evaluated our method using sources of sense-tagged corpus. In supervised learning, sense-tagged corpus was used to induce a classifier and then applied to classify test data. Our approach, however, was purely unsupervised and the sense tagged corpus was used to carry out an evaluation of the discovered sense groups. The way the tool used for processing clustering depends on the cluster mode one selects. For this study "Using training set" evaluation mode was selected. In this mode Weka first ignores the class attribute and generates the clustering and during the test phase it assigns classes to the clusters based on the majority value of the class attribute within each cluster. Based on the above technique its accuracy was used to measure how well it has been able to generalize the clustering result [16].

1.9 Organization of the Thesis

The thesis was organized into six chapters comprising Introduction, Literature review, the Tigrigna Language, Methodology, Experimentation and Discussion and Conclusion and

Recommendations. The first chapter gives the general introduction of the thesis. The second chapter presents reviews made on different literatures regarding WSD together with its approaches and different machine learning techniques. The third chapter reviews the Tigrigna writing system and ambiguities in the language. The fourth chapter discusses the methodology, which is composed of corpus preparation, system architecture and clustering and evaluation technique. The fifth chapter discusses the experimentation and discussion of the findings. Finally, chapter six deals with the conclusion and the recommendations drawn from the findings of the study.

CHAPTER TWO. LITERATURE REVIEW AND RELATED WORKS

2.1 Overview of Word Sense Disambiguation

In all the major languages around the world, there are a lot of words which denote meanings in different contexts. According to Solomon [16], Word Sense Disambiguation (WSD) is a key enabling technology that automatically chooses the intended sense of a word in context. It has been the focus of intensive research since the beginning of Natural Language Processing (NLP), and more recently it has been shown to be useful in several tasks such as Parsing, MT, IR, Question Answering, and WSD is considered to be a key step in order to approach language understanding beyond keyword matching.

As stated in [55], WSD serves as an intermediate step for computer science applications. Therefore, it has been a central problem since the earliest days of computational studies of natural language. Word Sense Disambiguation [51] is a technique to find the exact sense of an ambiguous word in a particular context. For example, Tigrigna word **‘ሓበለ’** read as “Habele” may have different senses as **“ዓይኑ ዓወረ”** read as “Aynu_Awere” which means (**unsighted**), “**ኣእመነ”** read as ‘’AEmene” which means (**convince**) etc. Such words with multiple senses are called ambiguous words and the process of finding the exact sense of an ambiguous word for a particular context is called Word Sense Disambiguation.

A normal humans have years of knowledge, experience, and an inborn capability to differentiate quickly the multiple senses of an ambiguous word in a particular context. A machine, however, has a much harder time finding the correct meaning and run only according to the instructions. It takes thousands of computations for even the simplest algorithms, which are not very accurate. Even so, many applications such as language translators are still available and sold today. Language translation relies heavily on word sense disambiguation. For this reason, many translated sentences do not make much sense. Solving word sense disambiguation would help with many applications such as language translation; due to that different rules are fed to the system to execute a particular task [54].

2.2 Brief History of Research on Word Sense Disambiguation

According to Kolte [56], WSD is one of the most challenging jobs in the research field of Natural Language Processing. Research work in this domain was started during the late 1940s. In 1949, Zipf proposed his "Law of Meaning" theory. This theory states that there exists a power-law relationship between the more frequent words and the less frequent words. The more frequent words have more senses than the less frequent words. In 1990s, three major developments occurred in the research fields of NLP: online dictionary Word Net [55] became available, the statistical methodologies were introduced in this domain, and Senseval began. The invention of Word Net brought a revolution in this research field because it was both programmatically accessible and hierarchically organized into word senses called synsets. Today, Word Net is used as an important online sense inventory in WSD research. Statistical and machine learning methods are also successfully used in the sense classification problems moreover, methods that are trained on manually sense-tagged corpora (i.e., supervised learning methods) have become the mainstream approach to WSD.

Alternatively, techniques have been proposed for discovering senses of words automatically from unannotated text. This task of unsupervised word sense induction (WSI) can be conceptualized as a clustering problem. To correctly identify all senses of polysemous words encountered in a corpus, words can be clustered according to their meanings and allowing multiple memberships [14].

In SensEval-3 (2004), a conclusion was reached that WSD in itself has reached a performance level, and no significant rise in the results obtained already is possible. It is since then, that people started thinking about new directions in which WSD research can go. In particular, in recent years there has been considerable growth in the areas of parallel bilingual corpora, and unsupervised corpus-based WSD. This study employs unsupervised WSD and attempts to draw upon the idea that unsupervised WSD is the way to go in future.

2.3 Application of Word Sense Disambiguation

The foremost field of application of WSD is Machine Translation, however it is used in near about all kinds of linguistic researches. Such as:-

Machine Translation (MT): WSD is required for MT [51], as a few words in every language have different translations based on the contexts of their use. For example, in the Tigrigna sentences, **"እቲ ስርዓት ተመንዩ።"** read as **"**Etiy sreAt temeniyu**",** this means 1. "**The system comes boring."** 2. "**The system predicts something."** the word "**ተመንዩ**" read as "temeniyu" carries different meanings which is a big issue during language translation.

Information Retrieval (IR): Resolving ambiguity in a query is the most vital issue in IR [16] system. As for example, a word "**ዓሊቡ"** reads as "Alibu" in a query may carry different meanings as **ኣብ ሓደ ቦታ ዓረፈ** read as "ab Hade bota Arifu" this means **arrived somewhere** and, **ኣግዘመ** read as "agzeme" which means **give over**. So, finding the exact sense of an ambiguous word in a particular question before finding its answer is the most vital issue in this regard.

Information Extraction (IE): Information Extraction is the task of automatic mining of structured information such as entities, relationships between entities, and attributes describing entities from unstructured sources [24]. This enables much richer forms of queries on the rich unstructured sources than possible with keyword searches alone. When structured and unstructured data co-exist, Information Extraction makes it possible to integrate the two types of sources and pose queries spanning them. IE tasks like named entity recognition (NER), acronym expansion (e.g., MP as Member of Parliament or Military Police), etc., can all be cast as disambiguation problems.

Speech Processing: Some words occur as several parts of speech. For example, in English language the word "blue" can be a noun, verb, or adjective. Each of these parts of speech has several senses, or definitions. The only way to tell the difference is to look at the context around the word. Knowing the part of speech would dramatically decrease the number of senses that are necessary, and it simplifies the problem. In addition, many researchers and competitions purposely focus on a single part of speech. Therefore, having an accurate part of speech tagger is very useful in word sense disambiguation [54].

There are many other NLP applications that could make use of WSD advancements in addition to those discussed above; such as Speech and Text processing, Content and theme Analysis, Semantic web endeavors, and so forth. This highlights that WSD is an intermediate task in a broad range of greater NLP applications, even if only implicitly.

2.4 Basic Methodological Approaches to Word Sense Disambiguation

Different approaches have been used through the evolution of WSD research. Many approaches have been proposed for assigning senses to words in context, while early attempts only served as models for toy systems. Word Sense Disambiguation Approaches are classified into Knowledge based approach and corpus-based approach [16].

2.4.1 Knowledge – Based Word Sense Disambiguation

Knowledge-based methods utilize lexical and semantic knowledge bases such as machine readable dictionaries (MRDs), thesauri, computational lexicons. Despite the efforts to automatically create knowledge bases, WordNet, the most widely used one, was created by hand. A brief introduction of WordNet is in order since almost all recent work on WSD has used WordNet in some way or another.

Generally four main types of knowledge-based methods (Algorithms) are used [56].

Table 2-1 Comparison of knowledge-based Algorithms

Lesk Algorithm	**Semantic Similarity**	**Selectional Preferences**	**Heuristic Method**
✓ depends on the overlap of the dictionary definitions of the words in a sentence	✓ Share common context and able to provide harmony to whole discourse. ✓ Various similarity measures are used to determine how much two words are semantically related.	✓ Improper word senses are omitted. ✓ Count how many times this kind of word pair occurs in the corpus with syntactic relation.	✓ The heuristics are evaluated from different linguistic properties to find the word sense.

As we have discussed in Table 2-1, Lesk algorithm (first machine readable dictionary based algorithm), Semantic Similarity (share common context and therefore the appropriate sense is chosen by those meanings), Selectional Preferences (improper word senses are omitted and only those senses are selected which have harmony with common sense rules) and, Heuristic Method

(evaluated from different linguistic properties to find the word sense) are the main types of algorithms built under the knowledge-based methods for word sense disambiguation.

2.4.2 Corpus – Based Approach

A major challenge facing WSD research is the ability to obtain a large amount of words with their different contexts. Corpus-based approaches came up with alternate solution to the challenge by obtaining information necessary for WSD directly from textual data which is called a corpus. A corpus provides a bank of samples which enable the development of numerical language models, and thus the use of corpora goes hand-in-hand with empirical methods [57].

Corpus-based approaches provide an alternative strategy to overcome the lexical acquisition bottleneck observed in knowledge-based approaches by giving information necessary for WSD directly from textual data. In this approach, the task of WSD is performed by training statistical or machine learning language models on a corpus.

If one chooses to work with a corpus-based approach, the possible means used for disambiguating senses of words are distributional information and context words. Distributional information about an ambiguous word is the frequency distribution of its senses. Collection information is obtained from context words which are found to the right and/or to the left of ambiguous words.

Corpus based approaches can be categorized into three sub classes based on the form of machine learning used for training [2]: Supervised Word Sense Disambiguation, unsupervised Word Sense Disambiguation and Bootstrapping Approach to WSD.

2.4.2.1 Supervised Word Sense Disambiguation

The supervised approaches applied to WSD systems use machine-learning technique from manually created sense-annotated data. Training set will be used for classifier to learn and this training set consist examples related to target word. These tags are manually created from dictionary. Basically this WSD algorithm gives well result than other approaches.

In supervised learning, a learning set is considered for the system to predict the meaning of ambiguous words using a few sentences having a specific meaning of the particular ambiguous words. Specific learning set is generated as a result for each instance of different meaning. A system finds the probable meaning of an ambiguous word for the

particular context based on defined learning set. In this method, learning set is created manually unable to generate fixed rules for specific system. Therefore predicted meaning of an ambiguous word in a given context can't be always detected. Supervised learning is capable to derive partial predicted result, if the learning set does not contain sufficient information for all possible senses of the ambiguous word. It shows the result, only if there is information in the predefined database [56].

Supervised learning approach having different methods and the most common are; Naïve Bayes, Decision Tree and, Decision List [15].

Table 2-2 Supervised WSD Methods

Naïve Bayes	Decision List	Decision Tree
✓ Is probabilistic classifier which is assigned the sense to target word that has maximum probability value	✓ Is based on some set of rules and decision is taken based on if-then -else conditions.	✓ Predication based method

Supervised learning approach having different methods such as Naïve Bayes, Decision Tree, and Decision List method (see table 2-2). Naïve Bayes algorithm is based on conditional probability, which calculate probabilities for each sense and assign the sense to target word that has maximum probability value. Decision List method is based on some set of rules. The decision is taken based on if-then-else conditions. Decision Tree [14] is a method which predication based method.

2.4.2.2 Semi – Supervised Word Sense Disambiguation

Semi-supervised methods can be defined as systems which train sense classifiers from annotated data with minimal or partial human supervision. Bootstrapping is the common approach of this kind. It works based on automatic bootstrapping of a corpus from a small number of manually tagged examples and on the use of monosemous relatives [21]. Bootstrapping is usually starts from few annotated data **A**, a large corpus of un-annotated data **U**, and one or more basic classifiers. As a result of iterative applications of a bootstrapping algorithm, the annotated corpus

A grows increasingly and the untagged data set **U** shrinks until some threshold is reached for the remaining examples in **U**.

Semi-supervised approaches are beneficial and effective since they use small labeled data and a bigger unlabeled training data. But there are some problems with them. These methods cannot obtain training data for senses in the lexicon that do not appear in the training data (unseen sense). Moreover, the major drawback with them is lack of a method for selecting optimal values for parameters like the pool size, the number of iterations and the number of most confident examples.

2.4.2.3 Unsupervised Word Sense Disambiguation

Unsupervised methods have the potential to overcome the knowledge acquisition bottleneck that is, the lack of large-scale resources manually annotated with word senses. These approaches to WSD are based on the idea that the same sense of a word will have similar neighboring words. They are able to induce word senses from input text by clustering word occurrences, and then classifying new occurrences into the induced clusters. They do not rely on labeled training text and, in their purest version, do not make use of any machine–readable resources like dictionaries, thesauri, ontologies, etc. However, the main disadvantage of fully unsupervised systems is that, as they do not exploit any dictionary, they cannot rely on a shared reference inventory of senses.

While WSD is typically identified as a sense labeling task, that is, the explicit assignment of a sense label to a target word, unsupervised WSD performs word sense discrimination, that is, it aims to divide "the occurrences of a word into a number of classes by determining for any two occurrences whether they belong to the same sense or not" [24]. Consequently, these methods may not discover clusters equivalent to the traditional senses in a dictionary sense inventory. For this reason, their evaluation is usually more difficult: in order to assess the quality of a sense cluster we should ask humans to look at the members of each cluster and determine the nature of the relationship that they all share (e.g., via questionnaires), or employ the clusters in end-to-end applications, thus measuring the quality of the former based on the performance of the latter. Admittedly, unsupervised WSD approaches have a different aim than supervised and knowledge-based methods, that is, that of identifying sense clusters compared to that of assigning sense labels. However, sense discrimination and sense labeling are both sub problems of the word

sense disambiguation task and are strictly related, to the point that the clusters produced can be used at a later stage to sense tag word occurrences.

Most of the time, supervised approaches are superior to unsupervised in terms of accuracy of automatic disambiguation when used on the same type of texts that the systems were trained on [16]. According Solomon [17] the cost of annotation preparing corpuses for supervised classification algorithm is high, because large effort is required during manual annotation.

Like the supervised learning, even the unsupervised WSD methods strive from the data sparseness problem, since enormous amounts of text are needed to ensure that all senses of a polysemy word are represented in the corpus.

2.5 Algorithms for Unsupervised WSD

According Jain [57], Clustering algorithms are generally categorized as partitional and hierarchical. The next section describes some common clustering algorithms. Here are general properties that characterize clustering algorithms.

Agglomerative vs. Divisive algorithms: In agglomerative algorithms (bottom-up approach), each element is initially its own cluster and then the most similar clusters are iteratively merged until we are left with one large cluster containing all elements or until a stopping condition is met. Where as, divisive algorithms (top-down approach) initially begin with a single all-encompassing cluster and iteratively split the clusters until each element belongs to its own cluster or until a stopping condition is met

Hard vs. Soft algorithms: Hard clustering algorithms assign each element to exactly one cluster on the other hand soft (fuzzy) algorithms may assign an element to multiple clusters. In soft clustering, a membership degree is associated with each element's assignment to a cluster

Deterministic vs. Stochastic algorithm: These types of searches mostly apply to partitional algorithms that optimize some clustering function. Stochastic algorithms use random searches of the feature space while deterministic algorithms do not.Throughout the next section; we use n to represent the number of elements that are to be clustered. When the number of clusters must be fixed by an input parameter, like in many partitional clustering algorithms, we refer to this number by K.

2.5.1 Hierarchical Algorithms

Hierarchical algorithms produce a nested partitioning of the data elements by merging or splitting clusters. Agglomerative algorithms iteratively merge clusters until an all-encompassing cluster is formed, while divisive algorithms iteratively split clusters until each element belongs to its own cluster. The merge and split decisions are based on the similarity metric. The resulting decomposition (tree of clusters) is called a dendrogram.

Figure 2-1 shows a possible dendrogram produced by an agglomerative hierarchical algorithm. At the topmost level of the dendrogram, we have a single cluster containing all elements. Using a similarity threshold, we can extract a clustering of the data by cutting the dendrogram according to this threshold.

Then, each connected component of the dendrogram forms a cluster. For example, assuming that the best clustering in the 2-dimensional space of Figure 2-1 consists of small tight clusters, the dotted line in (b) gives a good threshold for this data resulting in three clusters: The problem with any threshold is that on some data sets, a particular threshold will be good but on another data set, it will fail. For example, in Figure 2-1, if the similarity threshold was just a little higher, we would have five clusters with elements *C* and *D* in separate clusters.

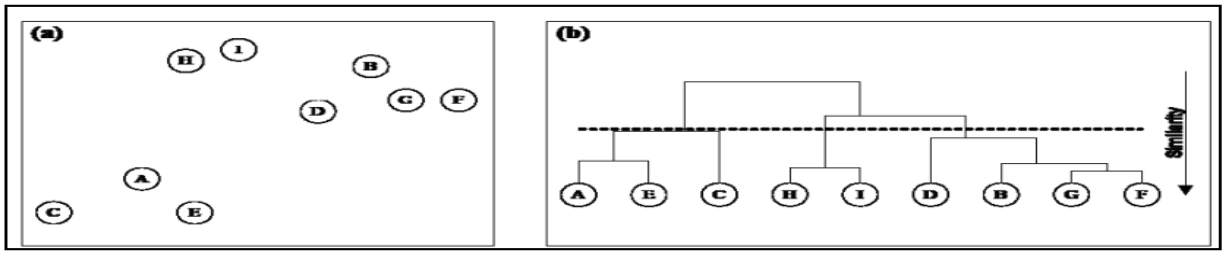


Figure 2-1 Dendrogram visualization of a hierarchical clustering result [16]

> *(a) Nine data points in 2-dimensional space; (b) the dendrogram produced by a hierarchical agglomerative clustering algorithm (the dotted line indicates a possible similarity threshold for selecting the final clustering).*

The dendrogram provides a visualization of how the algorithm produced its output. For example, if a particular output cluster is bad, the dendrogram provides a method of verifying how this bad cluster was formed. Hierarchical algorithms rigidly make merge and split decisions. If a particular decision is wrong, the algorithm will never go back and undo the decision. This makes

the algorithm more efficient than performing a combinatorial search of all possible decisions but it can never correct itself.

2.5.1.1 Agglomerative Clustering

Initially start with n clusters each containing a different element. In the final step, an all-encompassing cluster is created and the result is a dendrogram like the one in Figure 2-2. The different versions of agglomerative clustering differ in how they compute cluster similarity. The most common versions of agglomerative clustering algorithm are single-link, complete-link and average-link clustering. The complexity of these algorithms is $O(n^2 \log n)$ [21].

Single-link clustering: In single-link clustering the similarity between two clusters is the similarity between their most similar members (e.g. using the Euclidean distance) [16]. It is capable of discovering clusters of varying shapes like the clusters of Figure 2-3. However, single-link is not practical because it suffers from the chaining effect [45]. For example, in Figure 2-3 (b), single-link clustering generates an elongated cluster because of a bridge of elements connecting two clusters

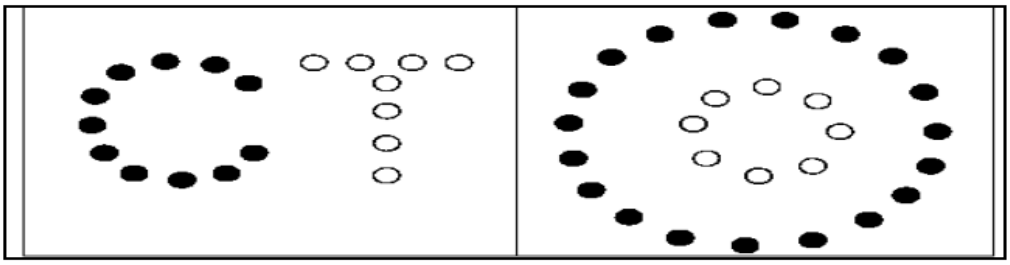


Figure 2-2 Clusters Discoverable using Single-link clustering [16]

Complete-link and average-link cannot discover these two clustering

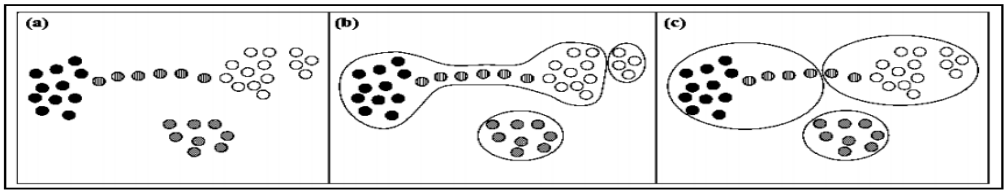


Figure 2-3 Chaining effect in Single-link Clustering [16]

(a) Data points in 2-dimentional space; (b) the clustering produced by single-link clustering; (c) the clustering produced by complete-link clustering. The proximity measure is the Euclidean distance

Complete-link clustering: In complete-link clustering, the similarity between two clusters is the similarity between their least similar members (e.g. using the Euclidean distance) [34]. Although complete-link clustering is not capable of discovering clusters like the two in Figure 2-3, it does not suffer from the chaining effect. Rather than producing straggly elongated clusters like single-link, complete-link generates compact clusters. Figure 2-3 (c) shows an example. Complete-link generates better clustering's than single-link in many applications [16]. Figure 2-4 illustrates the different computations for cluster similarity between single-link and complete-link.

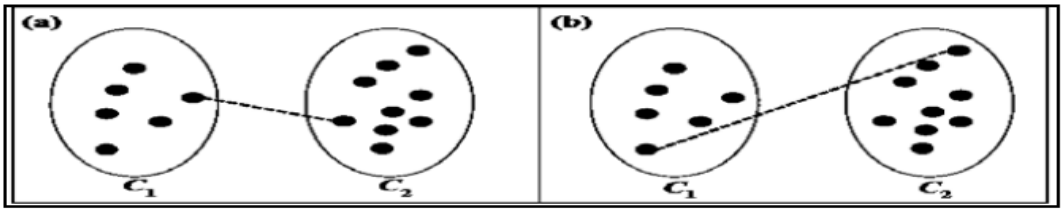


Figure 2-4 Single-links vs. Complete-link cluster Similarity [16]

C1 and C2 are two clusters in 2-dimensional space where their similarity is the similarity between the two elements joined by a dotted line for (a) the single-link algorithm and (b) the complete-link algorithm.

Average-link clustering: Average-link clustering produces similar clusters to completelink clustering except that it is less susceptible to outlier [58]. It computes the similarity between two clusters as the average similarity between all pairs of elements across clusters (e.g. using the Euclidean distance). Figure 2-5 shows snapshots of merge decisions comparing the three linkage algorithms on a 2-dimensional data set

Table 2-3 Difference between Single-Link, Complete-Link and, Average-link Clustering

Single-link clustering	Complete-link clustering	Average-link clustering
✓ similarity between two clusters ; the similarity between their most similar members ✓ capable of discovering clusters of varying shapes (Fig2-3) ✓ not practical because it suffers from the chaining effect(Fig 2-3 (b))	✓ similarity between two clusters; the similarity between their least similar members ✓ is not capable of discovering clusters (Fig2-3), ✓ it does not suffer from the chaining effect ✓ generates better clustering's than single-link in many applications	✓ produces similar clusters to complete link clustering except that it is less susceptible to outliers ✓ computes the similarity between two clusters as the average similarity

2.5.1.2 Divisive Clustering

Although it is not as common as agglomerative clustering [16]; Divisive clustering algorithms start with a single cluster containing all elements. Considering all possible splits of the cluster into two clusters gives $\boldsymbol{2^{(2n-1)} - 1}$ possibilities. Using a splitting heuristic to iteratively split the largest cluster, Divisive clustering algorithms has worst case time complexity $O(n^2 \log n)$.

Let the diameter of a cluster c be the similarity between the two least similar elements in

c. The algorithm is as follows:

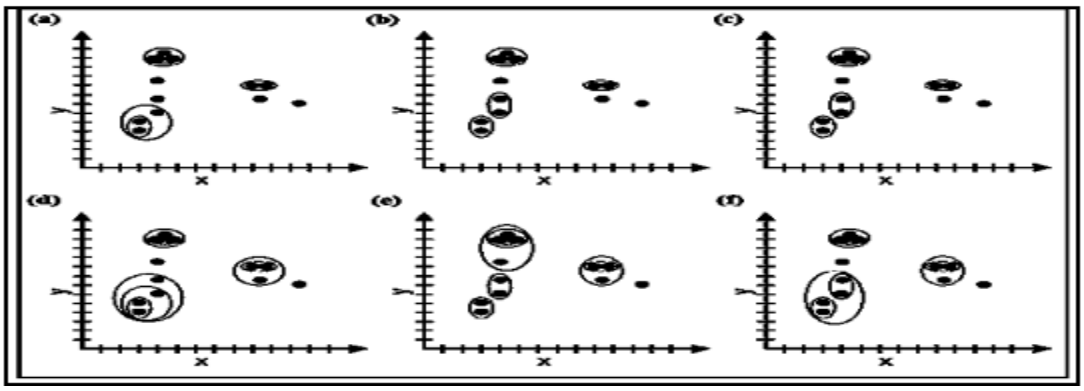


Figure 2-5 Single-link, complete-link and average-link clustering

Dotted ellipses denote previously merged clusters and solid ellipses denote newly merged clusters. (a),(b) and (c) illustrate the fifth merge decisions for single-link, complete-link and average-link respectively while (d), (e) and (f) illustrate the seventh merge decisions.

1. Initially start with a single cluster encompassing all elements;
2. Select l, the largest cluster or the cluster with highest diameter;
3. Find the element e in l that has the lowest average similarity to the other elements in L
4. e is the first element added to the splinter group while the other elements in l remain in the original group;
5. Find the element f in the original group that has highest average similarity with the splinter group;
6. If the average similarity of f with the splinter group is higher than its average similarity with the original group then assign f to the splinter group and go to Step 5; otherwise do nothing;
7. Repeats step 2-6 until each element belongs to its own cluster.

   ;

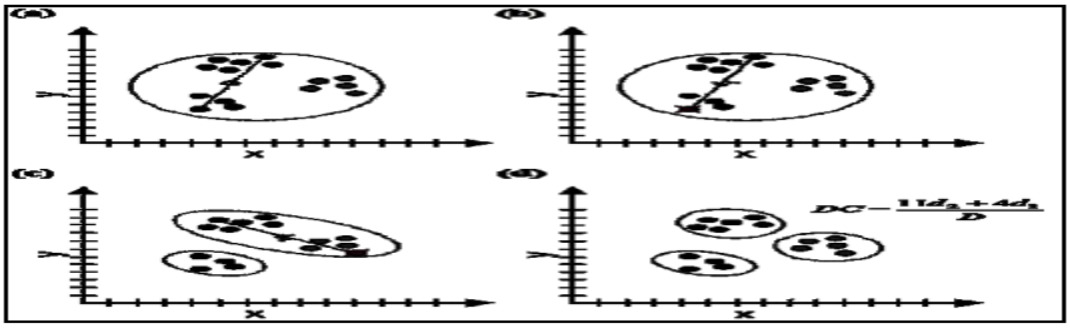


Figure 2-6 Divisive clustering

(a) the initial all-encompassing cluster with diameter D; (b) the first splinter group defined by the cross (d1 is the diameter of l from Step 2); (c) the result of the reassignment of elements to the splinter group after the first iteration - the new splinter group is defined by the cross (d2 is the diameter of the new l from Step 2); (d) the result of the reassignment of elements to the splinter group after the second iteration and the DC measure assuming that this is the final partitioning.

2.5.2 Partitional Algorithms

Partitional algorithms do not produce a nested series of partitions. Instead, they generate a single partitioning, often of predefined size *K*, by optimizing some criterion. A combinatorial search of all possible clustering's to find the optimal solution is clearly intractable. The algorithms are then typically run multiple times with different starting points. Partitional algorithms are not as versatile as hierarchical algorithms but they often offer more efficient running time [16].

2.5.2.1 K – Means

The most commonly used family of partitional algorithms is based on the *K*-means algorithm. *K*-means clustering is often used on large data sets since its complexity is linear in *n*, the number of elements to be clustered. It creates a partitioning such that the intra-cluster similarity is high and the inter-cluster similarity is low. *K*-means uses the concept of a centroid where a centroid represents the center of a cluster. A centroid is usually not an element from the cluster. Rather, it is a pseudo-element that represents the center of all other elements. Often the mean of the feature vectors of the elements within a cluster is used as that cluster's centroid. It is often difficult to define a centroid for categorical features.

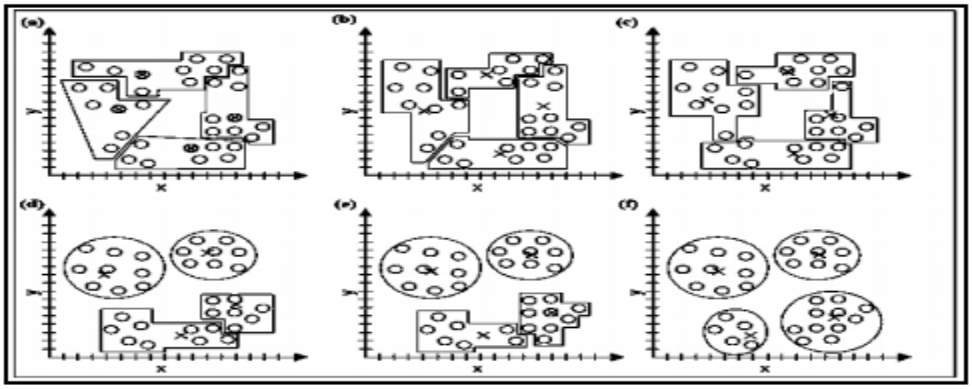


Figure 2-7 K-means clustering

The crosses represent cluster centroids and K=4. (a) The initial randomly selected centroids and the first cluster assignment; (b) – (f) the second to sixth iterations of K-means. After the sixth iteration, the element assignments do not change and the algorithm terminates.

K-means iteratively assigns each element to one of *K* clusters according to the centroid closest to it and recomputed the centroid of each cluster as the average of the cluster's elements. The following steps outline the algorithm for generating a set of *K* clusters [15]:

1. *Randomly select K elements as the initial centroids of the clusters;*
2. *Assign each element to a cluster according to the centroid closest to it;*
3. *Recomputed the centroid of each cluster as the average of the cluster's elements;*
4. *Repeat Steps 2-3 for T iterations or until a criterion converges, where T is a predetermined constant.*

2.5.2.2 Bisecting K – Means

Bisecting *K*-means [15], a divisive variation of *K*-means, begins with a set containing one all-encompassing cluster consisting of every element and iteratively picks the largest cluster in the set, splits it into two clusters and replaces it by the split clusters. Splitting a cluster consists of applying the *K*-means algorithm α times with *K*=2 and keeping the split that has the high average element-centroid similarity. Note here that $\alpha \neq T$. It is the whole *K*-means algorithm that is repeated α times. Each instantiation of *K*-means will have *T* iterations.

2.5.2.3 K – Medoids

The centroids constructed by *K*-means are sensitive to outliers, if there are many of them, since each element has a direct influence on the construction of the centroids. *K*-medoids [58, 59] is a family of algorithms that addresses this shortcoming. Instead of representing a cluster by its centroid, *K*-medoids uses one of the elements of the cluster as its representative. The algorithm is very similar to *K*-means. Initially, *K* random elements are chosen as the initial representative of the *K* clusters. In its iteration, the algorithm a representative element is replaced by a randomly chosen on representative element if the criterion (e.g. squared-error criterion) is improved.

2.5.3 Expectation Maximization Algorithm

Expectation maximization (EM) is a well-known algorithm used for clustering in the context of mixture models [60]. This method estimates missing parameters of probabilistic models. Generally, this is an optimization approach, which had given some initial approximation of the cluster parameters, iteratively performs two steps: first, the expectation step computes the values expected for the cluster probabilities, and second, the maximization step computes the distribution parameters and their likelihood given the data. It iterates until the parameters being optimized reach a fix point or until the log-likelihood function, which measures the quality of clustering, reaches its maximum.

The expectation maximization is an iterative estimation procedure in which a problem with missing data is present in a different form to make use of complete data estimation techniques [15]. In our work, the sense of an ambiguous word is represented by a feature whose value is missing.

In order to use the EM algorithm, the parametric form of the model representing the data must be known. In these experiments, we assume that the model structure is the Naive Bayes according [16]. In this model, all features are conditionally independent given the value of the classification feature, i.e., the sense of the ambiguous word. This assumption is based on the success of the Naive Bayes model when applied supervised word-sense disambiguation [49].

There are two potential problems when using the EM algorithm. First, it is computationally expensive and convergence can be slow for problems with large numbers of model parameters. To solve the above problem we used small data set for this study. Second, if the likelihood

function is very unbalanced it may always converge to a local maximum and not find the global maximum.

To simplify the discussion, we first briefly describe the EM algorithm. The algorithm is similar to the K-means procedure in that a set of parameters are re-computed until a desired convergence value is achieved. The parameters are re-computed until a desired convergence value is achieved. The finite mixtures model assumes all attributes to be independent random variables.

A mixture is a set of *N* probability distributions where each distribution represents a cluster. An individual instance is assigned a probability that it would have a certain set of attribute values given it was a member of a specific cluster. In the simplest case *N=2*, the probability distributes are assumed to be normal and data instances consist of a single real-valued attribute. Using the scenario, the job of the algorithm is to determine the value of five parameters, specifically as discussed by Solomon [15]:

1) The mean and standard deviation for cluster 1
2) The mean and standard deviation for cluster 2
3) The sampling probability *P* for cluster 1 (the probability for cluster 2 is *1-P*)

And the general procedure states as follow:

1) Guess initial values for the parameters.
2) Use the probability density function for a normal distribution to compute the cluster probability for each instance. In the case of a single independent variable with mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$, the formula is:

$$f(x) = \frac{1}{(\sqrt{2\pi\sigma})e^{\frac{-(x-\mu)2}{2\sigma 2}}} \quad \text{.....} \qquad \text{Eq. 2.1}$$

In the two-cluster case, we will have the two probability distribution formulas each having differing mean and standard deviation values.

3) Use the probability scores to re-estimate the five parameters.
4) Return to Step 2.

The algorithm terminates when a formula that measures cluster quality no longer shows significant increases. One measure of cluster quality is the likelihood that the data came from the

dataset determined by the clustering. The likelihood computation is simply the multiplication of the sum of the probabilities for each of the instances. With two clusters *A* and *B* containing instances *x1*, *x2*, *x3*,…,*xn* where *PA=PB=0.5* the computation is:

$$[.5P(x1|A) + .5(x1|B)][.5P(x2|A) + .5(x2|B)]... [.5P(xn|A) + .5(xn|B)] \quad \ldots\ldots \quad \text{Eq. 2.2}$$

Expectation maximization (EM) is a clustering algorithm that works based on partitioning methods. This algorithm is a memory efficient and easy to implement algorithm, with a profound probabilistic background. EM is widely used iterative algorithms for estimating model parameters in the presence of missing data, in our case: the missing data are the senses of the ambiguous words.

2.5.4 Hybrid Algorithms

Hybrid clustering algorithms are characterized as multi-phase algorithms that combine hierarchical and partitional techniques [58]. In this section, we present five algorithms: Buckshot, BIRCH, CURE, Rock and Chameleon.

2.5.4.1 *Buckshot*

According Solomon [16], Buckshot addresses the problem of randomly selecting initial centroids in K-means by combining it with average-link clustering. Buckshot first applies average-link to a random sample of n elements to generate K clusters. It then uses the centroids of the clusters as the initial K centroids of K-means clustering.

As the random sample-size approaches K, Buckshot degenerates to the K-means algorithm. The strict definition of the sample size makes Buckshot unsuitable for some situations. Suppose one wish to cluster 100,000 documents into 1000 newsgroup topics. Buckshot could generate at most 100,000 ≈ 316 initial centroids. The sample size counterbalances the quadratic running time of average-link to make Buckshot efficient: $O(K \times T \times n + n\log n)$. However, the algorithm can be run with any sample size as long as the speed of clustering is acceptable.

2.5.4.2 Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), is a two phase algorithm that uses a structure called a *CF*-tree to abstract the data yielding an efficient algorithm

[58]. A *CF*-tree is a compression of the data elements that attempts to preserve the inherent structure of the data. The two phases are:

1. *Construct a CF-tree by scanning through each element to be clustered;*
2. *Apply any clustering algorithm to cluster the leaf nodes of the CF-tree*

A *CF*-tree is a hierarchy of sets of clustering features. Given a sub cluster whose elements are represented by *m*-dimensional feature vectors, a clustering feature, *CF*, summarizes the information contained in the elements:

$CF = (N, \vec{LS}, \vec{SS}) \ldots$ Eq.2.3

Where N = number of elements in the sub cluster,

The first step of BIRCH has time complexity $O(n)$. As long as the chosen algorithm for step 2 is also linear (e.g. a partitional algorithm like *K*-means), BIRCH has overall time complexity $O(n)$, which is more efficient than Agglomerative and Divisive clustering algorithms. Because BIRCH uses a diameter parameter, it is not very good for discovering clusters that are not spherical. Another problem with BIRCH is that it is sensitive to the order in which the elements are scanned in Step 1 of the algorithm [15].

2.5.4.3 Clustering Using Representatives (CURE)

Single-link clustering has the advantage of being able to discover clusters of various shapes and sizes but it is not robust in the presence of outliers (i.e. the chaining effect). CURE (Clustering Using representatives), is similar in operation to single-link clustering but is more robust to outliers. Clusters are represented by a set of initially well scattered points that are shrunk towards the center of gravity of the cluster.

As satated in Guha [61], given a set of elements *X* to cluster, CURE initially selects a random sample of size *s* from *X*. The random sample is then partitioned into *p* partitions each with size *s*/*p* and then the partitions are partially clustered using an agglomerative hierarchical algorithm. Setting a high similarity threshold in aggregative analysis gives many small clusters. Clusters that grow too slowly are tagged as outliers and are eliminated. At this point, we have several small tight clusters and each is represented by the mean of its constituting elements (a centroid).

The time complexity of CURE is $O(n)$, making it efficient for large data sets. However, the algorithm is very sensitive to its input parameters: the shrinking factor α and the random sample size.

2.5.4.4 Robust Clustering Using Links (ROCK)

Robust Clustering using links (ROCK), is an algorithm for clustering binary and categorical data. Previous clustering methods that use a distance measure, such as the Euclidean distance between elements, are not suitable for binary and nominal data.

ROCK has worst-case time complexity of $O(n2 + nmmma + n2logn)$ where *mm* is the maximum number of neighbors and *ma* is the average number of neighbors. ROCK is a good algorithm for categorical data but its complexity makes it inefficient for large data sets.

2.5.4.5 Chameleon

CURE ignores the aggregate interconnectivity between two clusters while ROCK ignores the average closeness between clusters. Chameleon [54] combines the advantages of CURE and ROCK while employing dynamic modeling of clusters to improve clustering quality. Clusters are merged in Chameleon if they have high interconnectivity and closeness relative to each cluster's internal interconnectivity and closeness. Chameleon has been shown to produce higher quality clusters than CURE but it suffers from a worst case time complexity of $O(n^2)$.

2.6 Comparison among the Different Types of Approaches

As the test sets, sense inventories, machine readable dictionaries, knowledge resources which are required for different WSD algorithms are different, each algorithm has some advantage and disadvantages (see table 2-4).

Table 2-4 Comparison of WSD Approaches [10]

Approach	Advantage	Disadvantage
Knowledge-Based	✓ These algorithms give higher Precision.	✓ These algorithms are overlap based, so they suffer from overlap sparsity and performance depends on dictionary definitions.
Supervised	✓ These types of algorithms are better than the two approaches implementation perspective.	✓ These algorithms don't give satisfactory result for resource scarce language
Unsupervised	✓ There is no need of any sense inventory and sense annotated corpora in these approaches.	✓ These algorithms are difficult to implement and performance is always inferior to that of other two approaches.
Semi-Supervised	✓ needs only a few seeds instead of a large number of training examples unlike pure supervised approaches	✓ lack of a method for selecting optimal values for parameters like the pool size, the number of iterations

2.7 Related Works for Ethiopian Languages

Unlike English and other western languages, Ethiopian languages are less researched languages in the areas of Information Retrieval and Natural Languages Processing applications [17]. Recently four researches are done in the areas of IR and NLP for Ethiopian languages. Some of these researches are presented as follows.

Table 2-5 Related works for Ethiopian languages

Name	Title	Algorithm	Result
Udaya R.G. (2014)	✓ Supervised approach using WordNet for nepali language	✓ WordNet	✓ Accuracy 88.059
Ayan D., Sudeshna S., (2013)	✓ Un-Supervised Graph-based Approach for Bengali language	✓ Un-Supervised Graph based Approach	✓ Accuracy 60%
Prity B., (2013)	✓ Selectional Restriction for Hindi language	✓ Selectional Restriction	✓ Accuracy 66.92%
Teshome (1999)	✓ WSD based on semantic vector to improve the performance of an IR system modeled for Amharic legal texts	✓ semantic vector (develop his own Algorithm based on distributional hypothesis)	✓ Precision (58%) Recall (82%)
Solomon M. (2010)	✓ supervised machine learning approach to Amharic text	✓ Naïve Bayes supervised classifier algorithm	✓ Accuracy (70 – 80%)
Solomon A. (2011)	✓ unsupervised machine learning approach WSD to Amharic texts	✓ K-means, agglomerative single and Complete Link clustering algorithms.	✓ Accuracy (65.1 - 79.4 %)
Getahun (2012)	✓ WSD prototype model using semi-supervised machine leaning approach to Amharic texts	✓ combination of clustering and classification algorithms	✓ Accuracy (88.47%)
Hagerie W. (2013)	✓ Ensemble Classifiers Applied to Amharic WSD	✓ Ensemble classifiers	✓ Accuracy 78.75 - 80.46

2.8 Summary

A basic introduction to the field of WSD, application of WSD (e.g. MT, IR, IE,), and a survey of the major approaches to WSD such as Knowledge based, Corpus based approach has been presented. Further we explore Naïve Bayes, Decision Tree and, Decision List are the most common methods Supervised learning approach. Major categorized of clustering algorithm; Partitional and Hierarchical clustering as well as common clustering algorithms such as Agglomerative, divisive, hard, soft, deterministic and, stochastic algorithm. Comparison of approaches and related work in Ethiopia language also discussed.

A major challenge facing WSD research is the ability to obtain a large amount of words with their different contexts [16]. As a result this study used five selecting algorithms of Unsupervised WSD to get encouraged result on all of them than taking one algorithm, wastage of time to prepare word nets and finding labeled sources and tested on (see table 2-5).

CHAPTER THREE. WORD SENSE AMBIGUITY IN TIGRIGNA LANGUAGE

3.1 Overview of Tigrigna Language

The Semitic languages are the Afro-Asiatic language family. Arabic, Amharic, Tigrigna, Hebrew, and Aramaic are the most widely spoken Semitic languages today. There are different Semitic languages in Ethiopia. These are Amharic, Tigrigna, Gurage, Argobba, Gafat, Ge'ez [32].

Tigrigna is one of the Ethio-semantic languages. It is spoken in Tigray region and Eritrea. The original name of the language is TIGRAY. It is a written language with a certain amount of literature. As a northern language, Tigrigna had developed in the traditional home of Ethiopic civilization and continued to be spoken there. It is the most spoken language after Arabic and Amharic. The speakers of this language live in compact, densely populated, sedentary agricultural areas of the Tigray and Eritrean plateau [33]. This language has more than six million speakers worldwide [34].

As discussed in [33], Tigrigna alphabets are taken from Geez and its vocabularies are more or less similar with that of Geez. There is also an assumption given that Geez is the parent language of Tigrigna. And others assume that both languages are sister languages of the same origin, i.e. the Proto-Ethio-Semitic language. However, it is not yet settled the question the question of whether Geez is the parent language of Tigrigna or whether both languages derived from some Proto-Ethiopic language or languages. But some linguistic reasons have been given to show that Geez is not the Parent language of Tigrigna. One reason is that Geez does not have the prefix "**AI**" read as "ኣይ" or it's weakened from AY read as "ኣይ" to make negatives while others do except Tigre.

For more understanding summary of the relation of Geez and Tigre to Tigrigna is presented in figure 3-1.

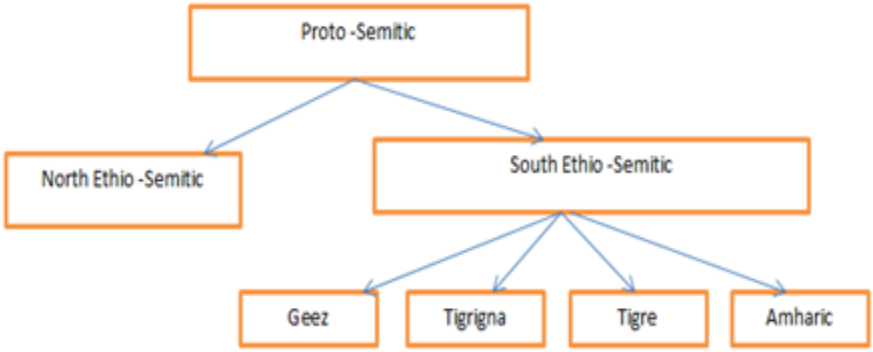


Figure 3-1 Ethio-Semitic Family [35]

To sum up, because of their geographical distribution, for they have a common origin, Tigrigna and Geez have the same alphabets and similarity of vocabularies as well. These four languages (Geez, Tigrigna, Tigre and Amharic) are said to be sister languages [35].

3.2 Tigrigna Writing Systems

The Ethiopic writing system is used to represent the four Semitic languages which originated from the Geez alphabet (the liturgical language of Ethiopian Orthodox Church). These are Ge'ez, Amharic, Gurage, and Tigrigna. These languages are limited to Ethiopia and Eritrea. Ge'ez is no more mother tongue of any person, but it still has a very significant role in the traditional language of literature and religion especially in Ethiopia Orthodox Church. Amharic and Tigrigna are closely related to each other [37].

As cited by Gebrehiwot [37], Ethiopic writing system is written from left to right. It does not make any distinction between upper and lower case letters and has no conventional cursive form. There are no systematic variations in the form of the symbol according to its position in the word. The Ethiopic system for Tigrigna language consists of alphabets, numbers, and punctuation marks.

Tigrigna language has its own characters (alphabets), punctuation marks, and number systems. As John [36] discussed, this language uses a special writing system called the "Ge'ez" ("Ethiopic") alphabet (or syllabary). The earliest inscription in Ge'ez is vowels. However, at some point in the distant past, the vowels of the language began to be written by way of small additions or modifications to the consonants preceding them. Thus, although Arabic and Hebrew are usually written without vowels, languages that use the Ge'ez alphabet are always use vowel.

The normal syllable in Tigrigna is considered to be a consonant followed by a vowel. If a consonant ends a syllable, the sixth, neutral vowel is used with it. Most consonants are written in seven slightly different forms corresponding to the traditional seven vowels. The system of vowels applied to all the consonants is shown in the alphabet chart, commonly called "he hu /ሀ ሁ…" [36].

3.2.1 Alphabets

Alphabets are sets of letters arranged in fixed orders of the language they used to write. They are also called phonemes which contain consonants and vowels. There are different alphabets representations in the world. The most alphabets representation is Latin or Roman alphabets which have been adapted by numerous languages. The Ethiopic writing systems have also their own writing systems. Similarly, Tigrigna has its own alphabets **ፊደል** read as "**Fidel**" and they are used for writing different documents of Tigrigna language. It has thirty-five base symbols with seven orders which represent seven vowels for each base symbol [38]. In addition to these symbols, Amharic has 44 additional variants for labialized consonants (plus vowel), that is, syllables involving consonants with lip-rounding, for example: mWa (ሟ), tWa (ቷ), kuWa (ኳ), ruWa (ሯ) etc.

As discussed by Gebrehiwot [37] Tigrigna writing system can be translated in Latin representation by finding a Latin letter with similar sound Tigrigna letter. For instance the Tigrigna letter 'ሀ'has a similar sound with Latin letters 'H'. As a result, the seven order of the letter 'ሀ' can be represented by combining the Latin letter 'H' with vowels as shown below in table 3-1:

Table 3-1 Sample of Tigrigna letter and their corresponding Latin letter

Order	1	2	3	4	5	6	7
Tigrigna	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
Equivalent Latin letter	He	Hu	Hi	Ha	Hie	H	Ho

3.2.2 Tigrigna Punctuation Marks

Punctuation marks are vital to know word demarcation for natural language processing as a result to that, the system or mashine understand the sentence easly. Most of the Tigrigna language punctuation marks are listed in Table 3-2 below.

Table 3-2 most commonly used punctuation marks with their English corresponding marks

Punctuation marks (Tigrigna)	**Meaning**	**Equivalent English Punctuation marks**
፡ (ክልተ ነጥቢ)	word separator	White space
። (አርባዕተ ነጥቢ)	End of sentence	.
፤ (ድርብ ሰረዝ)	Sentence connector	;
፣ (ነፀላ ሰረዝ)	List separator marks	,
፦ (ሃይፈን)	beginning of the list mark	:-
? (ሕቶ ምልክት)	End of question	?
! (ቃለ አጋንኖ)	End of an emphatic declaration, or command.	!
<< >> (ስርዓተ_ ነጥቢ)	quote some words or sentences taken from other	“ “
() (ቅንፍ)	Used to write a word synonym i.e. or	( )

3.2.3 Number System

Like Amharic numbering systems, Tigrigna number system uses Ge’ez numbering systems. It has twenty characters. They represent numbers from one to ten (፩-፲), twenty to ninety (፳-፺), hundred (፻) and thousand (፼) as shown below:

Table 3-3 numbering system

-	፩	፪	፫	፬	፭	፮	፯	፰	፱
0	1	2	3	4	5	6	7	8	9
፲	፳	፴	፵	፶	፷	፸	፹	፺	፻
10	20	30	40	50	60	70	80	90	100

3.3 Problem of Tigrigna Writing System

As discussed by in [4], there are a number of challenges in Tigrigna language for text processing.

- **Redundancy of some characters:** Sometimes more than one letter is used to represent similar sound in Tigrigna language. For instance, letters "ሀ" and "ኀ"; "ጸ" and "ፀ"; "ሰ" and "ሠ" have similar sounds. In the old literature of Tigrigna texts, the use of various forms of characters for the same sound has a problem in the process of feature preparation for the classifier learning. However, current literatures do not have such problems since it has only one letter for one sound. As a result, the alphabet "ኀ", "ጸ" and "ሠ" are no more in use in writing Tigrigna document.
- **Spelling variation of the same word:** Even though there are feasible problems of spelling variation in current literature of Tigrigna. A word may be translated by different persons using different spelling variation. For instance ሬድዮ/ራድዮ - 'radio'፣ ዝብላዕ/ልብላዕ - 'to be eat' and etc.
- **Abbreviation:** The abbreviations of Tigrigna words follow different formats. Sometime full stop '.' is used to abbreviate, while other time '/' symbol is used to abbreviate.

Generally, based on the linguistics view, each language in the world has its own writing system but it can be similar or difference. Since language can be seen from phonology, morphology, structure of grammar, syntax, semantics, and pragmatic. Like other languages, Tigrigna language has its own characters (alphabets), punctuation marks, and numbers systems as discussed above and morphological structure. As cited by Hailay [39] , due to its morphological richness,

Tigrigna exhibits the root and stem pattern morphological phenomenon. Because the morphological variation is the result of adding affixes to the root verbs or nouns to indicate number, gender, tense, possession, etc. It is necessary to understand the behavior of Tigrigna language stems and roots.

3.4 Tigrigna Morphology

As cited by Agustina [40] morphology is defined as a way of studying language or linguistics. It is about the way words are put together, their internal structure. Morphology is the part of linguistics that studies patterns of word formation within and across languages. Morphology tries to formulate rules that show the knowledge of the speakers of those languages. In addition, Martin [41] describes that morphology is the study of the internal structure of words. Morphological research aims to describe and explain the morphological patterns of human language.

Like every language has its own morphological process, Tigrigna language has its own word structure, word formation and affixes. And as stated in the previous sections, Tigrigna is a highly inflected language and has a complex morphology.

Like other Semitic languages, Tigrigna has rich verb morphology. Tigrigna verbs show different morph syntactic features based on the arrangement of consonant (C) -vowel (V) patterns. For example, the root ስብር ('sbr') /to break/ of pattern (CCC) has forms such as ሰበረ 'sebere' (CVCVCV) [1] in Active, ተሰበረ 'te-sebere'(te-CVCCV) in Passive.

As discussed in [6] Tigrigna is a highly inflected language and has a complex morphology. It exhibits the root and pattern morphological system. The Tigrigna root is a sequence of consonants and it represents the basic form for word formation. Tigrigna makes use of prefixing, suffixing and internal changes to form inflectional and derivational word forms. Tigrigna Nouns are inflected for gender, number, case and definiteness. For example, ሃገራት(hagerat) - countries, ተምሃራይ (temaharay) - male student, ተማሃሪት (temaharit) - female student. Tigrigna adjectives are inflected for gender and number. For example, ፀሊም (tselim), ፀለምቲ (tselemti) meaning 'black' (masculine), 'blacks' respectively.

[1] CVCVCV, C = Consonant, V = Vowel

3.5 Syntactic Structure of Tigrigna

The syntactic structure is formed by combining different words. Since Tigrigna word formation follows its own structure, the syntax of the language also exhibit a unique structure. The syntactic structure of Tigrigna is generally SOV (Subject-Object-Verb).The modifiers in such structure generally precede the word or the phrases they modify. For example, the Tigrigna equivalent for the English sentence "He played football" is **"ንሱ ኩዕሾ-እግሪ ተፃዊቱ** ("`nisu kuEsho-egri teTSAwitu") here, the subject is "`nisu" and the object is "kuEsho-egri" and the verb is "teTSAwitu". But, usually pronouns are omitted when used as a subject. For the above English sentence the usual way to say it in Tigrigna is, "kuEsho-egri teTSAwitu"; the pronoun "`nisu" (He) is implicit in the sentence and come part of the verb. In this case the verb indicates the pronoun that is left out in the sentence.

Question formation is the same as a declarative sentence except the usage of question mark at the end. That is to ask the question "did he go to the church?" in Tigrigna, the sentence "he went to church" is ended with question mark instead of the Tigrigna full stop (.). The Tigrigna equivalent is **"ንሱ ናብ ቤተ-ክርስትያን ከይዱ?"** ("`nisu nab bete-kRstYAn keydu?").

Sometimes, words that indicate the sentence is a question are added at the end of the sentence. In such cases the above question becomes "` nisu nab bete-kRstYAn keydu Dyu?". Here, the word "`Dyu" is added to indicate that it is a question.

3.6 Ambiguities in Tigrigna

As stated in [42] six types of ambiguity in Amharic there are also ambiguities in Tigrigna similar to: Phonological, Lexical, Structural, Referential, Semantic and Orthographic ambiguities. We now summarize each type of ambiguity and some of the examples are adopted from [42] by translating to Tigrigna.

3.6.1 Phonological Ambiguity

Phonological ambiguity is a result due to the sound used for the word from the placement of pause with in a structure which occurs in speech. It can be illustrated through the following example:

[እቲ ቆልዓ ተመን + እዩ]።

He is misled person.

In the above sentence '+' sign shows where the pause is. When the sentence is pronounced with pause it means "He is misled or deceive person" but the meaning differs if it is pronounced without pause .It will mean "He is boring or tedious person".

3.6.1.1 Lexical Ambiguity

Lexical ambiguity refers to a case in which either a lexical unit belongs to different part of-speech categories with different senses, or to a lexical unit for which there is more than one sense, while these different senses fall into the same part-of-speech category [42].There are three different factors that can cause lexical ambiguity which are: Categorical Ambiguity, Homonymy and Homophonous Affixes.

3.6.1.2 Categorical Ambiguity

Categorical ambiguity is a result from lexical elements which have the same phonological and homographic form but belongs to different word class. This will be more described using the following ambiguous word:

ሕዳር ሂቡኒ።

Hdar hibuni.

In the above example the underlined word "hidar" is ambiguous since it has both nominal and a verbal meaning. It has two interpretations:

i. Hi gave me on a month of "November". [With nominal meaning]

ii. He gave me unplowed farmland. [With verbal meaning]

3.6.1.3 Homonymy

Homonymy is the state of a given word's having the same phonological form (and possibly with same spelling) however with different meanings which will cause ambiguity.

For example, the word 'mbTSAH'(ምብዓሕ) have more than one meaning. These include: Sharing, and show sadness. In the sentence of 'TSbaH MbTSAH aleni' (ፅባሕ ምብዓሕ ኣለኒ): MbTSAH means show sadness because somebody has died. Where as in 'Ety genzeb meMaEre MbTSAH Aleni' (ኢቲ ገንዘብ መማዕረ ምብዓሕ ኣለኒ), the money should have share equally.

3.6.1.4 Homophonous Affixes

This ambiguity result when affixes serve as different word classes. The word can be morphologically analyzed in to three separate morphemes: The prefix, the root and suffix. Since the meaning of each morpheme remains the same across words, it creates similar words that are difficult to understand. For example, the definiteness marker /u-/ is phonetically identical with the third person genitive possessive suffix /-u/. Hence, nouns like wedU (ወዱ) are ambiguous between the defined reading 'the child' or the possessive reading 'his son'. The following example show how homophonous affixes cause ambiguity.

ሓሰሩ ነዲዱ።

HAseru nedidU.

The above sentence is ambiguous because the suffix /-u/ serves as a definite article or as a third person masculine marker. It has two different meanings:

i. The straw is burned-out. And

ii. His straw is burned.

3.6.2 Referential Ambiguity

This ambiguity arises when a pronoun has more than one possible antecedent, thus having as many reading as there are antecedents .The following sentence is an example of such ambiguity.

ሓጎስ ፈተና ስለዝሓለፈ ተሓጉሱ።

Hagos fetenA slezHAlefe teHaGUisu.

The above sentence has two different readings:

i. Hagos was pleased because he passed the exam.

ii. Some body was pleased because Hagos passed the exam.

3.6.3 Structural Ambiguity

Structural ambiguity resulted when a constituent of a structure has more than one possible position. By a structure we mean the way syntactic constituents are organized.

The following is an example of such ambiguity:

ናይ ዓረብ ታሪክ መምህር።

Nay Areb tArik meMhr.

of-Areb history teacher.

The above sentence can have two different interpretations:

i. a person who teaches Arab history

ii. an Arab who teaches history

It can be further illustrated using structural organization of the sub-constituent /tarik/ ‘history’. It is shown in the following labeled representation (N means Noun):

i. **[[**Nay-Areb tArik] [meMhr**]]]]**

N N N

“Nay Areb tArik meMhr ”

ii. [Nay-Areb [[tarik] [meMhr]]]

N N N

3.6.4 Semantic Ambiguity

Semantic ambiguity is caused by polysemy, idiomatic, and metaphorical constituents. The following sentence is an example Polysemy constituent which has multiple meanings.

ደስታ የለን።

DestA yelen.

The above sentences have two interpretations:

i. There is no happiness.

ii. Desta(name of a person) is absent.

Idioms refer to an expression that means something other than the literal meanings of its individual words. Idioms ambiguity can be illustrated using the following example:

ብዕራይ ወለደ።

bEray welede.

The literal meaning of the above example is “An ox gave birth to a calf” but the idiomatic expression refers to “impossible “to happen.

Metaphors have literal or non-literal (metaphoric) senses. The following is an example of metaphoric ambiguity:

ሓራስ ኣድጊ።

Haras Adgiy.

It has two different interpretations:

i. ’Generous act, kind’
ii. ’Donkey with new-born cubs’

3.6.5 Orthographic Ambiguity

Orthographic Ambiguity is resulted from geminate and non-geminate sounds. The ambiguity can be resolved using context. Though in some cases it might not be possible like the following example:

ኣገልግል ኢላትኒ።

Agelgil elatny.

The word “Agelgil“ is the cause of ambiguity. The sentence is ambiguous between the following meanings.

She ordered me to give service (“Agelgil”)

She asked me to bring a dish (“Agelgil”)

3.7 Summary

This research made a detailed study about Ambiguity which occurs across all levels of NLP. It is highly complex task to resolve these kinds of ambiguities, especially in upper levels of NLP. As discussed in [62], the meaning of a word, phrase, or sentence cannot be under stood in isolation and contextual knowledge is needed to interpret the meaning, pragmatic and world knowledge is required in higher levels. It is not easy to create a world model for disambiguation tasks. Linguistic tools and lexical resources are needed for the development of disambiguation techniques. Resourceless languages are lagging behind in these field compared to resourceful languages in implementation of these techniques. Finally, this study deals with lexical ambiguity of four Tigrigna words which was focused to be resolved using EM, KM, SL, CL and AL algorithms of unsupervised approach among the type of ambiguities that were explained above.

CHAPTER FOUR. CORPUS PREPARATION AND SYSTEM ARCHITECTURE

As discussed in the literature review part, one of the mechanisms to acquire sense examples is to use monolingual corpora of second language and translate the sense examples to the original language. For this study Tigrigna text corpus was used for acquisition of sense examples, based on which a total of 631 sense example sentences for the four ambiguous words are acquired from different Tigrigna text corpus.

4.1 Acquisition of Sense Examples

The Acquiring process started from translating the senses of the ambiguous words to their equivalent English words using Tigrigna-English Dictionary. Then using the translated English word sense example sentences containing the word is acquired from the English corpus .For example the Tigrigna ambiguous word **“መደብ (Medeb)”** has two senses that are “program”, “Traditional bed”. Using these two senses, sense example sentences are acquired. The English sentences were examined thoroughly to check that it correctly represents the right sense of the Tigrigna word. For instance, the Tigrigna word **“ገረብ (Gereb)”** has two senses “River” and “Jungle”. But the English word “Jungle” is ambiguous by itself. It has six senses in English WordNet. But only sentences that have “Jungle” senses that match to the word “**ገረብ**” are selected.

As discussed in [33] the accuracy of classifiers degrade significantly when the training and testing samples have different distributions for the senses. In this study we tried to use a balanced distribution of senses for the ambiguous words to maximize performance when enough sense examples are available. On average, about 100 example sentences were acquired for each sense of ambiguous words with the exception of two senses on which enough example senses were not acquired from the corpus. The distributions of senses are summarized in table 4.1

Table 4-1 Senses of selected ambiguous words

Ambiguous Words	Tigrigna Representation	Sense 1,2,3,4 respectively		Number of Sentences for each word
Medeb	መደብ	Plan	56	156
		Traditional bed	61	
		Grouping	39	
Hademe	ሃደመ	Running	46	89
		Building house	43	
Halefe	ሓለፈ	Pass	110	270
		Pass away	56	
		Promote	59	
		Boss	45	
kebere	ክበረ	Expensive	51	116
		Respectfully	65	
Total				**631**

4.2 Corpus

The researcher has used different sources of sample text for the development of the word sense disambiguate and the experiments. Like other Ethiopian languages Tigrigna language also have no well organized corpus available like BNC collections for the English language [29]. The corpus prepared for this study consists of 631 sentences collected from the following sources:

Table 4-2 Corpus preparation from different Tigrigna sites

Source sites	Number of sentence
• in www.dmtsiweyane.com	205
• www.woyengazeta.org	95
• Tigray youth site: www.tigrayyouth.org	17
• www.wuraynamagazine.com	175
• http://tigrigna.voanews.com/a/	30
• http://www.daerona.com/news.html	29
• Tigrigna bible	80

The document collected from the above sources covered topics such as politics, economy, religion, science, medical, sport and love. The topics are helpful to represent the different morphological variation of words. And these source documents have been used for stop word list compilation, affix compilation and to test the algorithm. The test data was compiled from the document randomly for each word to check the stemmer from different angles of varieties of words.

4.3 System Architecture

The architecture of the system is depicted in Figure 4-1.The system takes sentences that contain the ambiguous words as an input .The sentences preprocessed to make them suitable for further processing. And ambiguity checker check the preprocessed sentence. Then the unsupervised algorithm (clustering algorithm) builds model from the training set and evaluate the built model and displays performance valuation of the model. The detailed explanations of the processes presented in the next subsections.

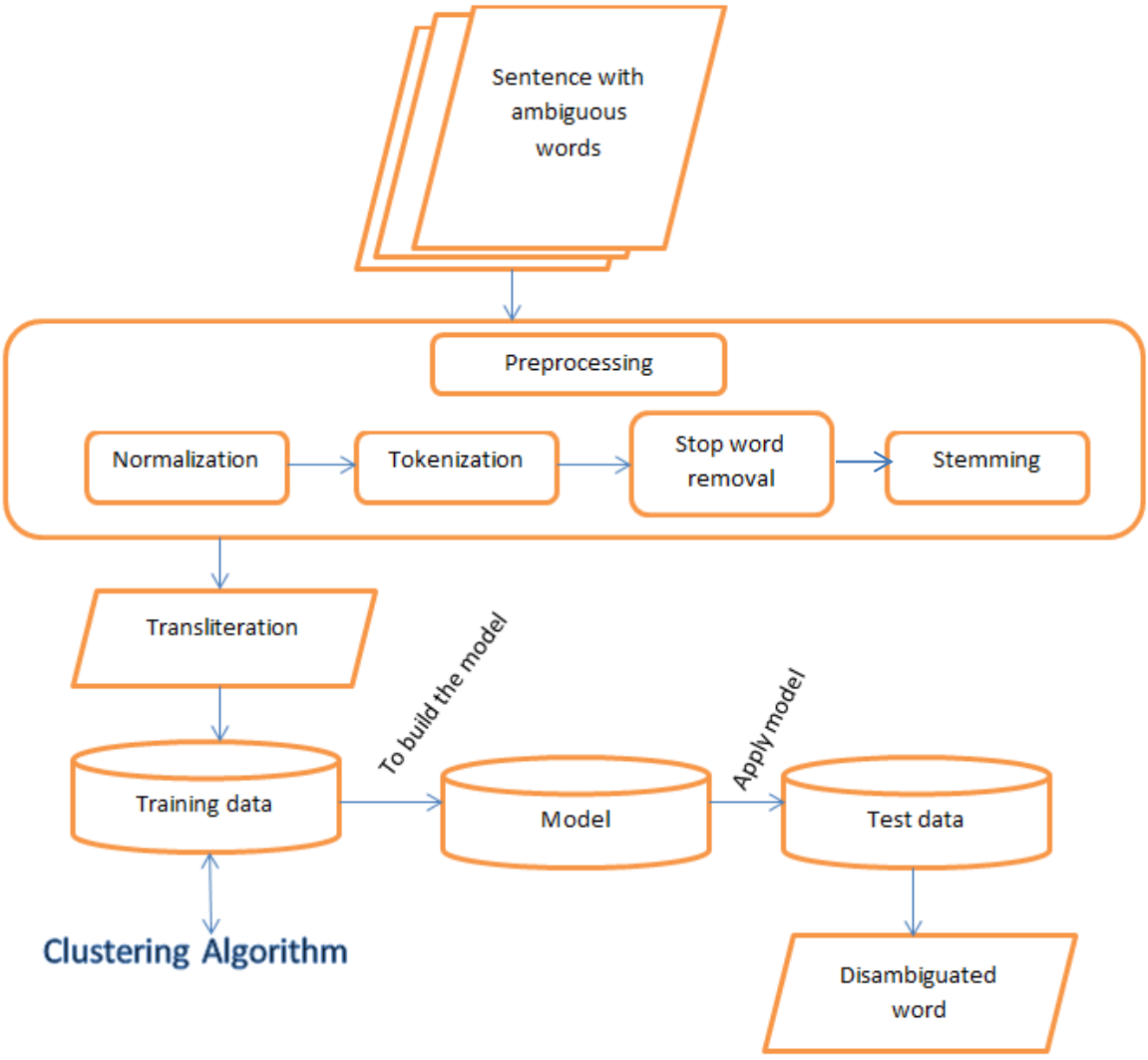


Figure 4-1 Unsupervised Word Sense Disambiguation System Architecture for Tigrigna

4.3.1 Document Preprocessing Techniques and Algorithms

As discussed in [29], preprocessing is the most important part of all text processing. It must ensure that the source text be presented to NLP in a form usable for it. For example, NLP programs usually need their input to be tokenized, i.e. text elements usually word forms or sentences are identified and placed on separate lines of the input [33]. Due to this reason, preprocessing is very important for this study to make the collected corpus usable for natural language processing. In order to preprocess the collected Tigrigna corpus, different text preprocessing techniques such as normalization, tokenization and stop word removal are used.

4.3.2 Normalization

In Tigrigna writing system there are many abbreviations (short) words. These short words can be single or compound short forms of Tigrigna language words abbreviated by either slash (/) or period (.). In order to make the corpus usable for stemming process, the short words should be expanding to their expanded form. And the Tigrigna language normalizer expands Tigrigna language short forms that are separated by either period or slash like ሕ.ወ፟ረ፟ት or ሕ/ወ፟ረ፟ት= ሕንጣሎ ወ፟ረ፟ት which is the name of district in southern Eastern Tigray about 65 k.m from Mekelle. In this study the short forms are represented by Unicode values taken from the Ethiopic Unicode table.

The compound Tigrigna language short form words are two types: with or without space between the two words, but they need a space after they are expanded. For example, ቤት ት/ቲ = ቤት ትምህርቲ which means ‘a school’ is a compound short word have a space between the two words ቤት and ት/ቲ. However, the compound short word ሓ.ዘመን have no space but after expanding it needs a space between the two words and expanded as ‘ሓዲሽ ዘመን’ which means ‘New Year’. Therefore, by considering such conditions, we have implemented Tigrigna language short form words expander.

Table 4-3 Normalization algorithm [29]

1. Open file and short word list
2. Normalize the document

```
while(not end of corpus file)
        read word
    if the word is in short form word list
        expand the word
    end if
end while
```

4. Close file

4.3.3 Tokenization

Tokenization is the process of splitting a text into words (tokens) so as to get context words for disambiguation purpose [17]. And also define as the process of splitting a string into a list of pieces, or tokens. This means that a given string is splitting into a list of tokens for the purpose of making usable to natural language processing [39].

In this study, all punctuation marks (see table 3-2) and numbers (see table 3-3) in Tigrigna language are removed from the text before the data is processed. And words are taken as tokens. The punctuation marks are converted into space. And space is used as a word separation. Hence, if a sequence of characters is followed by space, that sequence is identified as a word.

Table 4-4 Tekonization Algorithm

1. Open document
2. Tokenize word document

   While (not end of corpus file)

   read word

   if the word is in tokens list

   split the word

   end if

   end while
3. Close file

4.3.4 Stop Word Removal

According to [44], stop words are common in every natural language. These are common words that carry less significant meaning than the keywords in a document. Since stop words consume memory space and decrease the efficiency of the IR system by slowing the searching speed, search engines usually remove these words from a keyword phrase to return the most

relevant result. Therefore, removing the stop words saves space and speeds up the searches of documents.

In this study, a java program (using net beans 8, and jdk 1.8) has written to remove Tigrigna stop word using list consisting of pronouns (ንሱ፣ንሳ፣ንሳቶም፣ ንሳተን), prepositions (ብላዕሊ፣ብታሕቲይ), conjunctions (ነገር ግን፣ ወይካዓ) and articles (እዚ፣እቲ ሓደ) [29].

The Algorithm designed for removing Tigrigna stop words in this study looks like the following:

Table 4-5 Stop Word List Algorithm

1. open corpus and stop word list
2. while(not end of the corpus file)

   read the word

   If the word is found in stop words list then

   remove the word

   End if

   End while
3. close file

4.4 Stemming Tigrigna Word variants

4.4.1 Stemming

As states in [29], the Tigrigna language makes use of prefixing, suffixing and infixing to create inflectional and derivational word form. A stemming is a technique used to reduce words to their root form, by removing derivational and inflectional affixes (prefixes, infixes and suffixes) and will lead to significant improvement in WSD systems and in information retrieval tasks. A word's stem is its most elementary form which may or may not have a semantic interpretation. In documents written in natural language, it is hard to retrieve relevant information. Since the Languages are characterized by various morphological variants of words,

this leads to mismatch vocabulary. In applications using stemming, documents are represented by stems rather than by the original words [45].

Table 4-6 Stemmer Algorithm

1. open corpus, exception list and stop word list
2. While not end of corpus file is reached do

    Read terms

        For each term in the file

        If term starts with prefix

            If term not in exception file list then

                Remove prefix

            End If

        End If

        If term ends with suffix

            If term not in exception file list then

                Remove suffix

            End If

        End if

    End for

3. End while
4. Close files

4.4.2 Prefix Stripping

This step takes the output document from irregular word handling step. In prefix striping, it considers the cases when a prefix is not a real prefix. Therefor, before removing the prefix it checks whether the word is found in prefix exception list or not. If the word is in prefix list it

returns the word. If the word is not in prefix exception list and its length is greater than three characters, it checks the prefix of the word is found in prefix list or not. If the prefix is in prefix list, it removes the prefix. This process repeats up to the end of the file.

4.4.3 Suffix Stripping

This step takes the output document from prefix striping step. It considers the cases when a suffix is not a real suffix. Then before removing the suffix it checks whether the word is found in suffix exception list or not. If the word is in suffix exception list it returns the word. If the word is not in suffix exception list and its length is greater than three characters, it checks the suffix of the word is found in suffix list or not. If the suffix is in suffix list, it removes the suffix. This process repeats up to the end of the file.

4.4.4 Infix Stripping

This take the output document from suffix removal step and check whether infix is found in the word iteratively up to the end of the file. If infix is found within a word it removes the infix. Finally, it changes the last order (ፊደል) of each word to 6th order (‘sads’) and generates the stemmed document. The stemmer changes the last alphabet of each word to 6th order (‘sads’) because almost all Tigrigna stems end with 6th order (‘sads’) according to [33].

4.5 Transliteration

In addition to the above preprocessing, the Tigrigna documents need to be transliterated. For computational efficiency and simplicity of processing, transliteration of Tigrigna documents was used. Transliteration is the representation of the characters of one language by corresponding characters of another language (in this research Latin alphabets are used for transliteration). It enables easy, unambiguous and consistent communication of documents.

The transliteration of the Tigrigna corpus was conducted by using System for Ethiopic representation in ASCII (SERA) [46]. By transliterating the entire text, it was possible to have normalized the representation of words in different forms to one common form.

SERA, is case sensitive, i.e., upper and lower cases of the English alphabet representing different symbols in the Amharic language alphabet the same to that for Tigrigna language alphabet because of both are semantic languages . Therefore, we can represent simply the Tigrigna alphabets to its corresponding Latin alphabets.

Input: ናይ ኢትዮጵያ ተቓዋሚ ዉዱብ ኣብ ከተማ ባህር ዳር ከከያዶ ሒዙዎ ዝነበረ መደብ ንኻሊእ ግዜ ክልውጥ ብምምሕዳር እታ ከተማ ምውሳኑ ተቓዊሙ።

Output: Nay Ethiopia teQawami wudub ab ketema bahr dar kekaYdo Hizwo znebere medeb nkalE gzE klwet bmmHdar eta ketema mwusanu teQawimu.

4.5.1 Context Extraction

Context in WSD refers to the words surrounding the ambiguous words which are used to decide the meaning of the ambiguous word.

For instance, the sentence: "ነቲ ትማሊ ዝቐረበ መደረ ፕረዚደንት ኢሳያስ ኣፈወርቂ ብበሊሕ ነቒፎሞ ኣለዉ።"

After stop word removal and stemming, it will be "ትማል ቐረብ መደር በሊሕ ነቒፎም።"

The contexts are words surrounding the ambiguous word "ቐረበ" which are (መደረ፣ በሊሕ፣ ነቒፎሞ). In this study, the contexts of the ambiguous words are extracted using the algorithm of stemming of the corpus.

4.6 Training and Testing Datasets

Once all the necessary preprocessing tasks were done on the corpus, training and evaluating the selected algorithms followed. As discussed in [47], for this study there is no need to split the data into training and test sets for evaluation because of unsupervised nature of clustering algorithms. Table 4-4 shows the description of attributes in the data set. In the table, Rcontext (i) and Lcontext (i) refer to (ten words to the left and right) the words that surrounds the ambiguous word to the right and left respectively, where i ϵ (1, 2, …, 10), the target word holds the ambiguous word and Word class takes the senses of the ambiguous word, but the word classes are not practically used for experimentation (clustering senses) rather, they were used for evaluation of clustering assignments. If the i^{th} left or right word from the target word doesn't exist, an empty value was assigned to mean that there is no context. We have found that, the longest sentences in the corpus constitute a maximum of ten words to the left and the right of the ambiguous word. So we used 10 words to the left and the right of the ambiguous word as possible contexts. We were further explained using the following example which was extracted from the corpus.

LC **RC**

1) ካብ ውሽጣዊ ኣታዊና በጀትና ንሸፍን ዝብል **መደብ** መንግስቲ እውን ክትግበር ዝኽእል ፀጋታትካ ብዝግባእ ምጥቃም እንትከኣል እዩ።

LC **RC**

Kab wushtawi atawina beJetna nshfin zibl **medeb** mengsti ewin kitgber zikEl tsegAtatka bzgbaE mtkam entK'al iyu.

In the above example, the target word "**medeb**" which is the ambiguous word and its word class is "plan" (see appendix 9) that is its sense in this context. LC refers to the left context whereas RC refers to right context. There are six left contexts and nine right contexts surrounding the target word which are labeled as is shown in the example. But there is no one right context (10) and there are no four left contexts (7, 8, 9 and 10) which were assigned as empty. Note the word classes are used for evaluation of clusters assignment.

Table 0-7 Description of Attributes used for this study [17]

No	Attribute	Description	Value
1	Lcontext(i)	Used to hold the i^{th} left word from the ambiguous word	Any word in the corpus
2	Rcontext(i)	Used to hold the i^{th} right word from the ambiguous word	Any word in the corpus
3	Target word	Holds the ambiguous word	ambiguous word
4	Word class	Hold the label of the target word	Different sense of the ambiguous word

4.7 Evaluation Technique

As discussed in [48] evaluation of the clustering result can be done in many ways. Some of them are based on external criteria, i.e., the comparison of the resulting clustering solution with some preexisting categories that were created manually. On the other hand, one can use internal criteria without resorting to gold standard clustering. Beside to that as stated in [49] the most

important drawback of evaluation using internal criteria is that good score does not always correspond to good results of clustering in a given application. For the purpose of this study, annotated corpus was used for evaluation. The problem with WSD is its small size. Therefore there is a risk of not capturing all of the peculiarities and biases of some large corpora in WSD.

We evaluate our method using sources of sense-tagged corpus. In supervised learning sense-tagged corpus is used to induce a classifier that is then applied to classify test data. Our approach, however, purely unsupervised and the sense tagged corpus was used to carry out an evaluation of the discovered sense groups. The way Weka evaluates the clustering's depends on the cluster mode you select. For this study, using training set evaluation mode was selected in current implementation of Weka 3.8.1 package in order to satisfy our evaluation method. In this mode Weka first ignores the class attribute and generates the clustering. Then during the test phase it assigns classes to the clusters, based on the majority value of the class attribute within each cluster. Then it computes the clustering error, based on this assignment and also shows the corresponding confusion matrix [50]. Based on the above technique its prediction accuracy was used to measure how well it has been able to generalize the clustering result.

4.8 Selected Algorithms for Testing

For this study, we have selected five clustering algorithms for experimentation with the existing implementation in Weka 3.8.1 package but we tried to choose algorithms representing a few different approaches or techniques (that is, partitional, hierarchical and probabilistic approach) to the problem of clustering.

As a task of WSD is a contextual one, the cluster contexts (text snippets) containing ambiguous word. From the context some real-valued features are extracted. So the context is a vector of features **V** in high dimensional space. The feature vector comprises of attribute-value pairs, where the attributes are those contextual clues important for clustering.

First, we started with simple K-means algorithms, which represent simple, hard and flat clustering methods [16]. This algorithm has its drawbacks in terms of computational complexity, i.e., O (k (n−k) ^{2}), where n is number of contexts to cluster and k is number of centroid. This approach was applied in our experiments, as we have relatively small datasets.

Second, we choose agglomerative single, average and complete link clustering algorithms as representative family of hierarchical clustering algorithms. Last but not least, we test also the Expectation Maximization algorithms also known as the EM which is probabilistic clustering algorithms. It solves the maximization problem containing hidden (incomplete) information by an iterative approach [16]. In the setting of WSD, incomplete data means that the contextual features are not directly associated with word senses. The WSD is equivalent to choosing a sense that maximizes the conditional probability, P (X|Y, Θ). And also its performance is still highly competitive.

4.9 Summary

In this chapter, the design of the WSD system for Tigrigna was presented and discussed. More over the researcher includes those not explored before by researchers such as handling irregular words which gives another meaning for the ambiguity words and stemming them, and improving removal of prefix, infix, and suffix words in detail than worked before. Using the design, the process of corpus preparation, training and testing data sets, experimental evaluation technique and selected clustering algorithms for experimentation were illustrated in detail. The next chapter deals with the experimentation and discussion on the results of the experiment.

CHAPTER FIVE. EXPERIMENTATION AND DISCUSSION

5.1 Overview

Unsupervised word sense disambiguation was selected to use a set of unlabeled data and automatically find sense distinctions for this study. Usually those methods involve some form of clustering. For WSD, learning the unsupervised machine learning procedures not required providing explicit sense labels, where each data set example is described by a feature vector within each target word and in their sense label. The feature vector comprises of attribute-value pairs, where the attributes are those contextual clues important for clustering. For these studies four ambiguous words namely **መደብ** read as "**medeb**" which means (Program, Traditional bed and Grouping), **ሓለፈ** read as "**halefe**" which means (Pass, Boss, promote and pass away), **ሃደመ** read as "**hademe**" which means (Running and Building house) and **ከበረ** read as "**kebere**" which means (Expensive and Respecting) are trained for each ambiguous word with their corresponding data sets that are defined in Chapter four.

There is a need to split the data into training and test sets for evaluation because of unsupervised nature of clustering algorithms [16], [17]. These features are extracted from text in the following process. First a text window surrounding ambiguous word of ±10 segments (word) is constructed. Then the occurrence of a target word is noted in a feature vector for every dimension corresponds to different word. Then using Euclidian distance function, which is default in Weka package, can be used for measuring similarities between contexts. In this chapter the experimental procedures with the analysis of the experiment results was presented.

5.2 Experimentation Procedure

There were four main steps involved in the implementation stage using simple K means, EM and agglomerative single, complete and average link clustering algorithms that are implemented in Weka 3.8.1 Package. The details of the processes involved within each stage are described below.

Implementation steps

- ✓ **Step 1: Pre-processing -** This involved reading the data into the program, cleaning the data, removing stop-words and stemming (see appendix 11).

- ✓ **Step 2: Generating an Arff File** - This stage involved generating an Arff files for use with Weka. The features that were encoded into the Arff file were specified by the user (see appendix 13).
- ✓ **Step 3: Using Weka for Clustering** - This stage involved loading the Arff file into Weka, Using a variety of clustering algorithms , and showing the bar graph of different classes (see appendix 12).
- ✓ **Step 4: Evaluation of Clusters** - This section involved running the evaluation program to gain the accuracy of the clusters. Such as:-
    1) Check to what extent stemming and stop word removal of Tigrigna words in the corpus will affect the accuracy of unsupervised Tigrigna WSD (see table 5.1).
    2) Explore the outcome of dissimilar context sizes on disambiguation accuracy for Tigrigna ambiguous word. Here, different training data sets was organized for each ambiguous words, where the contextual information was gained from 1-left and 1-right to 10-left and 10-right following adjacent words are ready for each ambiguous word.

5.3 Discussion of Results

Experiment I: The effect of stemming on the accuracy of the result (selected ambiguous word)

Stemming has been found to give a significant upgrading on performance of WSD for morphologically rich languages (fig 5-1). This investigation is performed to test whether this applies to unsupervised WSD for Tigrigna. And “Use training set” evaluation mode was selected to test the experiments. During the test phase it assigned training set, based on the majority value of the class attribute within each cluster. Then it computed the clustering error. From these its prediction accuracy was used to measure how well it has been able to generalize from the training data to evaluate the model with the same labeled data set.

The result of this experiment after stemming is presented as follows in figure 5-1:

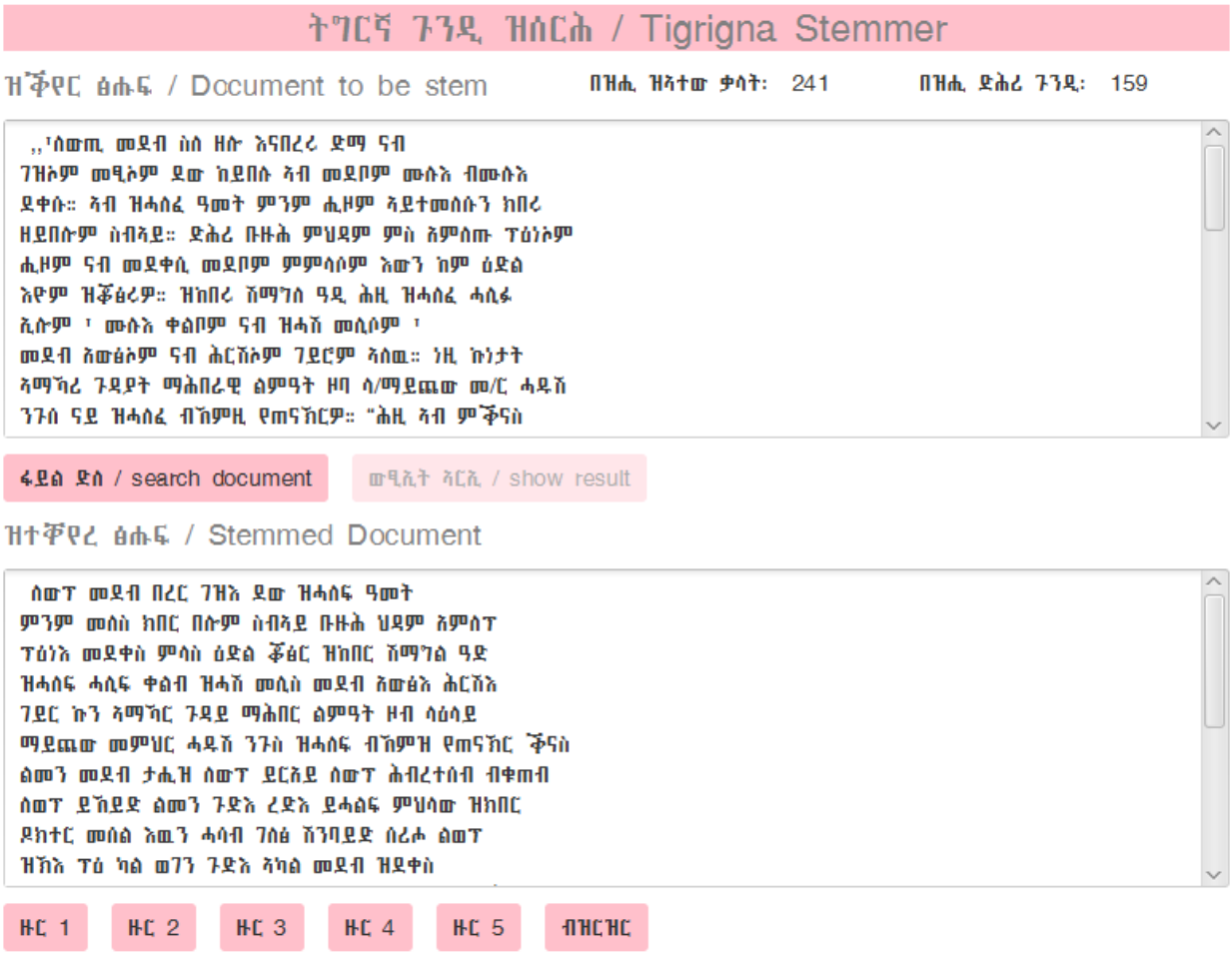


Figure 5-1 Stemming input (241) and output screen (159)

Figure 5-1 shows, the data set of containing ambiguous words were 241 before preprocessing as an input and after preprocessing gives 159 stemmed words as an output. How ever, there were some challenges in preprocessing for irregular words to get the correct meaning after stemming because they change their meaning to meaningless or other meaning (eg. Before stemming ኣጓላት read as "agualat" which means "girls" after stemm the word gives "ጓላ" read as "guala" which is meanigless) and in some words there were also a problem not correctly remove their insufix due to that their stemming is not correct. Over all the stemming process enhances to get a high result of accuracy on our data set using the five selected algorithms.

Table 5-1 Effect of stemming on accuracy of the classifier using training set cluster

Ambiguous Word	Accuracy											
	EM		**K-means**		**Single Link**		**Complete Link**		**Average Link**		**Average**	
	Before	After	Before	After	Before	After	Before	After	Before	After	Before	After
Medeb	58.2	64.3	69.1	75.3	52.5	59.3	50	53.1	57.3	56.8	57.42	61.76
Halefe	57.3	60.1	66.6	67	57.5	60.3	51.1	51	65.7	69	59.64	61.48
Hademe	60.3	59.7	63	68	56.4	59.8	62.7	63	64.3	61.9	61.34	62.48
Kebere	54.8	61	53.6	56.7	61.3	63	65	66.6	70.8	71.5	61.1	63.76
Average Improvement											**59.87**	**62.37**

As indicated in table 5-1 for all words, stemming improved the accuracy of all ambiguous words in simple k-means, Expectation Maximization and agglomerative single, complete and average clustering algorithms besides to that the accuracy of average improvement all selecting algorithms is improve. But, for those ambiguous words ‘Halefe’ in Complete Link and ‘Medeb’ and ‘Hademe’ in Average link agglomerative clustering algorithms, stemming doesn’t improve the accuracy of the algorithm. On average in all of each clustering algorithms, stemming improved the accuracy of the algorithms (see table 5-1). The reason behind the enhanced accuracy might be that, stemming brings variants of a word into their common stem. This minimizes the consideration of the variants of a word as different word by WSD model.

WSD models determine the meaning of a word by learning the pattern of surrounding words. If stemming is done, the variant of a word is taken as the same pattern, which will improve the accuracy of the algorithm for instance, before stemming, surrounding words ዝተበልዐ፡፣በሊ፟ም፣ በሊ፟ን and በሊዓቶ፟ም would be assumed as different but, basically they are the variants of the same word “ብልዕ“. After stemming, these words are taken as the same pattern. Therefore, in subsequent experiments the stemmed dataset was used as it enhanced the performance of the models.

Experiment II: Determining optimal context window

In English a standard two-word window on either side of the ambiguous word is found to be enough for disambiguation [16], [17]. But, for Amharic using supervised machine learning approach and unsupervised machine learning approach were found three-word window is enough. But, this has not been established for Tigrigna unsupervised WSD. For this study experiments were carried out ten times for each classifier to determine an average optimal window size from one-one window to ten-ten window on both side of the ambiguous word.

Table 5-2 Summery of Window Size experiment for Medeb

Window Size	Accuracy					
	Medeb					
	EM	K-means	Single Link	Complete Link	Average Link	Average
1-1	74.1	73	50.5	67	58.7	64.66
2-2	**83.1**	53	48.2	**73.3**	60.7	63.66
3-3	80.1	55	67.6	67.1	67.3	67.42
4-4	80.3	58	**74.1**	67.6	**70.8**	**70.16**
5-5	70.2	54	52.6	65	58.8	60.17
6-6	76.3	52	73.2	63.3	68.2	66.6
7-7	68.2	65	44.3	67	55.6	60.02
8-8	67	**77.3**	55.7	**73.3**	64.5	67.56
9-9	67.2	77.1	55.6	65.5	60.5	65.18
10-10	73.3	73	53.8	67.9	60.8	65.76

Table 5-3 Summery of Window Size experiment for Halefe

Window Size	Accuracy					
	Halefe					
	EM	K-means	Single Link	Complete Link	Average Link	Average
1-1	80.1	55	67.6	67.1	67.3	67.42
2-2	**80.3**	**77.5**	**74.1**	67.6	**73.7**	**74.64**
3-3	70	54	52.6	65	58.8	60.08
4-4	76	52	73.2	63.3	68.2	66.54
5-5	68	65	44.3	67	55.6	59.98
6-6	67	77.2	55.7	**73.3**	64.5	67.54
7-7	67	58	55.6	65.5	60.5	61.32
8-8	73.3	73	53.8	67.9	60.8	65.76
9-9	74.1	73	50.5	67	58.7	64.66
10-10	79.2	53	48.2	73	60.6	62.8

As shown in table 5-2 and 5-3 for the ambiguous word medeb in EM and CL and for halefe in EM, KM, SL and AL the maximum accuracy was achieved on two-two word window size. Where as, for medeb in K-means (KM) and Complete Link (CL) the highest accuracy was attained on eight – eight and for Single Link (SL) and Average Link (AL) the highest accuracy was attained on four –four respectively and for halefe in CL the highest accuracy was attained on six - six. How ever, the over all average accuracy of medeb on four – four (i.e. 70.16) and halefe on two – two (i.e. 74.64) was achieved.

Table 5-4 Summery of Window Size experiment for Hademe

Window Size	Accuracy					
	Hademe					
	EM	K-means	Single Link	Complete Link	Average Link	Average
1-1	76.1	52	52.6	63.3	57.9	60.38
2-2	68.2	**73.7**	**73.2**	67	**70.1**	**70.44**
3-3	67.7	69.8	44.3	**73.3**	58.8	62.78
4-4	67.6	70.1	55.7	65.5	60.6	63.9
5-5	73.4	65	55.6	67.9	61.7	64.72
6-6	74.2	63.3	53.8	67	60.4	63.74
7-7	**80.1**	53	50.5	**73**	61.7	63.66
8-8	80	55	58.2	67.1	62.6	64.56
9-9	70.3	58	67.6	67.6	67.6	66.22
10-10	76.7	54	72.1	65	68.5	67.26

As shown in table 5-4 and 5-5 for the ambiguous word Hademe in KM, SL, AL and Kebere in CL the maximum accuracy were achieved in two-two word window. Where as, for Hademe EM in seven - seven and CL in three – three the highest accuracy was achieved. For Kebere EM in four - four, KM in one – one, SL and AV in nine –nine the highest accuracy was attained. However, the over all average accuracy of Hademe on two - two (i.e. 70.44) and Kebere on eight - eight (i.e. 70.04) was achieved.

Table 5-5 Summery of Window Size experiment for Kebere

Window Size	Accuracy					
	Kebere					
	EM	K-means	Single Link	Complete Link	Average Link	Average
1-1	67	67.2	73.2	65.5	69.3	68.44
2-2	73	73	54.3	**67.9**	61.1	65.86
3-3	74.9	55	45.7	67	56.3	59.78
4-4	**83.3**	58	45.6	73	59.3	63.84
5-5	79	54	53.8	67.1	60.4	62.86
6-6	79.8	52	50.5	67.6	59	61.78
7-7	70	65	48.2	65	56.6	60.96
8-8	76.9	**77**	67.6	63.3	65.4	**70.04**
9-9	68.8	65.5	**74.1**	67	**70.5**	69.18
10-10	67.9	63.3	52.6	64.3	58.4	61.3

As shown in the table 5-2 to 5-5, all the four ambiguous words and for each clustering algorithms, the result agreed with the findings in other language that the nearest words surrounding the ambiguous word give more disambiguation information than words far from the ambiguous word[16].

Since in all ambiguous words, Window size of 2-2 was considered to be effective for Simple K means, EM, agglomerative SL and CL clustering algorithms the same to that their average accuracy.

5.4 Summary

In this chapter the experimental procedures together with presentation and discussion of three experiments were covered. The first experiment was showed effectively stemming on the data sets of containing ambiguous words. In successive using the stemmed data set the experiment was showed significantly improved the accuracy of the result before steeming the average b (see table 5.1). Moreover the sample stemming of the large corpus were found 14289 stemmed words from 20009 unprocessed words (see appendix 11). Finally experiments "using training set" mode, an experiment also carried out to determine optimal window size for the four ambiguous words. As a result by taking their average accuracy of each word window from one – one up to ten –ten of all algorithms; then two - two word window was found most favorable window size for clustering algorithms. During the experiment we face some Challenges on organizing word window size ten to the right and ten to the lefet, handling of missing windows and their replacement to weka understandable format. After handling correctly those all challenges we get the final accuracy of unsupervised Tigrigna WSD algorithms were achieved within the range of 63.66 – 70.14 % for word Medeb, 59.98 – 74.64% for word Halefe, 60.38 – 70.44% for Hademe and 59.78 – 70.04% for Kebere which was encouraging that compared to Unsupervised and Supervised Amharic WSD reported by [16]. As a result, EM algorithm is the best for Tigrigna WSD as compare to the ohers selected algorithm; as the reason missing data is present in different form to make use of complete data estimation.

CHAPTER SIX. CONCLUSION AND RECOMMENDATION

6.1 Conclusions

The overall focus of this research is Word Sense Disambiguation (WSD) which addresses the problem of automatically deciding the correct sense of an ambiguous word based on its surrounding context's. WSD is essential tool for NLP and IR applications .WSD is considered to be one of the most challenging of all NLP research areas due to its reliance on a varied range of linguistic and statistical knowledge.

The problem of WSD is addressed for Tigrigna which is one of less studied language. Though Tigrigna has many ambiguous words due to knowledge acquisition bottleneck, four ambiguous words are selected and clustering for each ambiguous word has been built. The words are **መደብ** read as "**medeb**", **ሓለፈ** read as "**halefe**", **ከበረ** read as "**kebere**", **ሃደመ** read as "**hademe**"

The most popular approaches to WSD rely on supervised machine learning methods, where a machine learning classifier is required to be trained on manually labeled training instances, to generate a classifier model that can be used to classify future instances. But manually labeling (annotation) training instances is too costly and time consuming. According Getahun [2] identified that the cost of annotation preparing corpuses for supervised classification algorithm is higher, because large effort is required during manual annotation.

In this study, unsupervised machine learning approach using five selected algorithms were used; these are Simple k means, EM and agglomerative single, average and complete link clustering algorithms. This method avoids the problem of knowledge acquisition bottleneck, that is, lack of large-scale resources manually annotated with word senses. This approach to WSD has been based on the idea that the same sense of a word will have similar neighboring words. They are able to induce word senses from input text by clustering word occurrences, and then classifying new occurrences into the induced clusters. They do not rely on labeled training text and, do not make use of any machine–readable resources like dictionaries, thesauri, ontology, etc. However, the main disadvantage of fully unsupervised systems is that, as they do not exploit any dictionary, they cannot rely on a shared reference inventory of senses and the result accuracy is less than that of supervised WSD method [16].

Based on selected algorithms, experiments on Weka 3.8.1 package, we conclude that simple k means, EM and CL clustering algorithms were achieved higher accuracy on the task of WSD for

selected ambiguous word in corpus. We have achieved accuracy within the range of 52 to 77.5% for Simple k-means, 67 to 83.3 for EM, 45.6 to 74.1 for Single, 65 to 73.3 for AL and 65 to 73.3 for Complete Link clustering algorithms for the four ambiguous words. But the worst results were in SL clustering algorithm in 4 – 4 window size due to time complexity in computing of minimum distance.

We also found that, stemming of Tigrigna words in the corpus enhanced the accuracy of the algorithms. The accuracy was increased after stemming was applied to words in the corpus.

For Tigrigna unsupervised WSD, there is no standard optimal context window size before, which refers to the number of surrounding words that are sufficient for extracting useful disambiguation. Based on this study, we have found that two-word window on each side of the ambiguous word was enough for disambiguation for Simple k means and EM, Single, Complete clustering algorithm but not in Average clustering algorithm.

Finally, the best unsupervised Tigrigna WSD algorithm is EM with accuracy of 67 to 83.3 for kebere, CL 65 to 73.3 for halefe, EM 67.6 to 80.1 for hademe and EM 67 to 83.1 for medeb respectively; because, this algorithm estimates missing parametrs of probabilistic models. These results of four ambiguous words were encouraging as usually unsupervised approaches do not rely on labeled training [16].

In conclusion, the chosen methodology, unsupervised machine learning approach for Tigrigna word sense disambiguation has been justified in terms of its theoretical foundations as well as the results obtained in our experiments for selected Tigrigna Ambiguous words. However, we face challenges in this research work. Such as, lack of prepared data set, incompatability of Tigrigna scripit directly by weka due to that it takes a lot of time to transliterate into latin alphabets each word, handling irregular words and handling those words didn’t stemm correctly, because of that they change their original meaning especially if their infix is not removed properly. This mybe the result of using insufficient stemmer algorithm but, for a feature the clustering algorithm can be efficient if the stemmer algorithm is enhanced than we use in this study and also if the irregular words are handled correctly in the exception as well.

6.2 Recommendation

Researches in word sense disambiguation require a variety of linguistic resources like thesaurus, WordNet, machine readable dictionaries, effective Tigrigna language stemmer, correctly identified of irregular words and machine translation software in which we faced a significant challenge as Tigrigna lacks those resources. The other challenge was lack of sense annotated data which makes the study to be using unsupervised approach for four ambiguous words. In this study we have only experimented with unsupervised machine learning approach but there are other approaches which performed well for WSD in other language. Therefore; the following recommendations are identified for further work in order to enhance WSD to Tigrigna texts, and the result useful in development of other NLP applications in Tigrigna:

1) Researches in WSD for other languages use linguistic resources like Thesaurus, Lexicon like WordNet, machine readable dictionaries and machine translation software. In this study, we faced a significant challenge as Tigrigna lacks those resources. Taking into account their contribution to WSD and other researches concerned institutions should develop these resources.
2) For other language a standard sense annotated data are available for WSD research and also for testing a WSD systems. We don't have such data for Tigrigna language which makes the study to be limited for four ambiguous words. So, there need to be an initiative to prepare the data for WSD research.
3) Future research directions for WSD in Tigrigna include:
    - ✓ Extending this experimentation using Supervised and unsupervised WSD for other ambiguous words in addition to those covered in the research
    - ✓ This study experiment only five clustering algorithms that are implemented in Weka 3.8.1 package. But other algorithms like Clustering by Committee (CBC), Growing Hierarchical Self-Organizing Map (GHSOM) and Graph-based algorithms has been tested as they are used and found to yield impressive result for other language[16], [25].
    - ✓ In addition to corpus based approach, there are also knowledge based and hybrid approach (combination of knowledge base and corpus based approach) which are used for WSD for other language and found a good result [6], [34]. These approaches need to be investigated for Tigrigna as well.

- ✓ A research should be conducted using bootstrapping approach which is required little training data and yields a very high performance. For example, an evaluation of Yarowsky's bootstrapping algorithm leads to very high performance over 90% accuracy on a small-scale data set [1]. This approach overcome the main problems of supervision and the data scarcity problem specially lack of annotated data like Tigrigna.
- ✓ Researchers can be study on how the word compounding and affixes affect Tigrigna words and their stems for WSD.
- ✓ Develop an efficient full-fledged Dialect based Tigrigna language stemmer by including all dialect based irregular and exceptional words. And apply the stemmer in WSD from Tigrigna texts.
- ✓ In this study some Tigrigna short forms are considered. But to enhance the accuracy of WSD including of all short words in Tigrigna language, and handles Tigrigna short forms that contain more than one slash or period should be studied.
- ✓ Due to the reason that there are Tigrigna words that are dialect irregular words, short forms, the Tigrigna stemmer must be researched in order to bring these irregular words into their stem.
- ✓ By incorporating necessary elements, the stemmer can also be used as a component for developing other computational tools like morphological analyzer, parser, machine translation, word frequency counting and other natural language applications.
- ✓ Tigrigna language is a morphologically reach language and needs more morphological knowledge of the language. Therefore, researchers can enhance the stemmer by creating team with Tigrigna experts formally for full effective stemming.

REFERENCE

[1]. Björn G. and, Lars A. “Experiences with Developing Language Processing Tools and Corpora for Amharic”, Kista, Sweden

[2]. Getahun W., “A Word Sense Disambiguation Model for Amharic Words using Semi-Supervised Learning Paradigm” A Peer-reviewed Official International Journal of Wollega University, vol. 3, 147-155 November 2014, Ethiopia

[3]. Gerard E. “Machine Learning Techniques for Word Sense Disambiguation” Barcelona, May 22, 2006

[4]. S.K.Jayanthi and S. Prema ” Word Sense Disambiguation in Web Content Mining Using Brill‘s Tagger Technique”, International Journal of Computer and Electrical Engineering, Vol. 3, No. 3, June 2011

[5]. Mohammad N. “A State of the Art of Word Sense Induction: A Way Towards Word Sense Disambiguation for Under-Resourced Languages”, Univ. Grenoble Alpes, (n.d)

[6]. Omer O. and, Yoshiki M. “Stemming Tigrinya Words for Information Retrieval” Nagaoka University of Technology, Nagaoka, Japan

[7]. David J. “An Analysis and Comparison of Predominant Word SenseDisambiguation Algorithms” Faculty of Computing, Health and Science Edith Cowan University, 24th June 2011

[8]. Arindam R. *et’al* “Knowledge Based Approaches to Nepali Word Sense Disambiguation” Department of Computer Science, Assam University, Silchar, International Journal on Natural Language Computing(IJNLC) Vol. 3, No.3, June 2014

[9]. Philip R. and, David Y. ”A Perspective on Word Sense Disambiguation Methods and Their Evaluation” Dept. of Linguistics/UMIACS University of Maryland, Dept. of Computer Science/CLSP Johns Hopkins University

[10]. Ping C. and, David B. “A Fully Unsupervised Word Sense Disambiguation Method UsingDependency Knowledge”, Dept. of Computer and Math. SciencesUniversity of Houston-Downtown, The 2009 Annual Conference of the North AmericanChapter of the ACL, pages 28–36,Boulder, Colorado, June 2009

[11]. A. Eneko and, G. Rigau, "Word sense using conceptual density" In Proceedings of the 16th International Conference on Computational Linguistics (COLING), Copenhagen, 1996 ndon, vol. A247, pp. 529–551, April 1996.

[12]. S.Weiss,Learning to Disambiguate.Information Storage and Retrieval,.9:p:33-41.1973

[13]. Kelly F. and J.S.Philip, computer recognition of English Word Sense North Holland, Amsterdam 1975

[14]. Michael D. "A Survey of Techniques for Unsupervised Word Sense Induction" Language Technologies Institute Carnegie Mellon University, December 4, 2009

[15]. Alok R., and Diganta S. "Word Sense Disambiguation: A Survey", International Journal of Control Theory and Computer Modeling (IJCTCM) Vol.5, No.3, July 2015

[16]. Solomon A. "Unsupervised Machine Learning Approach For Word Sense Disambiguation To Amharic Words" Adiss Ababa university, June, 2011

[17]. Solomon M. "Word Sense Disambiguation For Amharic Text: A Machine Learning Approach", Adiss Ababa university, June, 2010

[18]. Clara et'al, "A spreading-activation theory of semantic processing", Psychological Review, 1975. Vol 86(6).

[19]. Anderson, J. R., Language, Memory, and Thought.1976: Hillsdale, NJ.

[20]. Roberto, N. , "Word Sense Disambiguation: A Survey. ACM Computing Surveys", 2009. Vol 41(2).

[21]. Yarowsky, D. unsupervised word sense disambiguation rivaling supervised methods. in Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics. 1995. Cambridge, M.A.

[22]. Hearst and Marti A., Noun homograph disambiguation using local context in large corpora, in presented at Proceedings of the 7th Annual Conference of the University of Waterloo Centre for the New OED and Text Research,19 99: Oxford, United Kingdom.

[23]. Xinglong, W. and John, C. Word Sense Disambiguation Using Automatically Translated Sense Examples. In Proceedings of Human Language Technology

Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP). 2005. Association for Computational Linguistics.

[24]. Schutze **"**Automatic word sense discrimination", journal computational linguistics – special issue on word sense disambiguation, volume 24 issue 1, USA , march 1998

[25]. Hiroyuki K. and Yasutsugu M. "Unsupervised Word Sense Disambiguation Using Bilingual Comparable Corpora", Central Research Laboratory, Hitachi, Ltd, Tokyo, Japan

[26]. Hagerie W. "Ensemble Classifiers Applied to Amharic Word Sense Disambiguation", Addis Ababa University, June 2013.

[27]. Bors K. *et'al* "Word Sense Disambiguation for Information Retrieval" Artificial Intelligence Laboratory Massachusetts Institute of Technology, Cambridge, Massachusetts 022139

[28]. Krister "Word sense discovery and disambigua tion", University of Helsink, 2005

[29]. Tsegay G., "Towards performance improvement for Tigrigna language stemmer" university of Gondar, Ethiopia, 2016

[30]. Atelach A. and L.Asker. "An Amharic Stemmer: Reducing Words to their Citation Forms",M.S Thesis, Department of Computer and Systems Sciences, Stockholm University, Sweden, 2010

[31]. Abhishek F. *et'al* "An Approach for Word Sense Disambiguation using modified Naïve Bayes Classifier", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 4, Nagpur , India, April 2014

[32]. Hammond, J., A Chronicle of the Revolution in Tigray region of Ethiopia, 1999, Red Sea Press, Eritrea.

[33]. ትግራይ, ማ.ባ., መፅናዕትታት ቀዳማይ ሲምፖዚየም ቋንቋ ትግርኛ 1990, መቀለ: ብርሃን እና ሰላም ማተሚያ ቤት(Birhan ena selam press).

[34]. Leslau, W., Documents Tigrigna, 1998, Paris: Libraire CKlincksieck.

[35]. Bender, M.L., Language in Ethiopia 1976, London Oxford University Press.

[36]. John M, Tigrigna Grammar. 1996, Lawrencevine, New Jersey: Red See Press.

[37]. Gebrehiwot A, a two-step approach for tigrigna text categorization, in Department of Information Science 2011, Addis Ababa.

[38]. Leslau, W., Documents Tigrigna, 1998, Paris: Libraire CKlincksieck.

[39]. Hailay Beyene, Design and Development of Tigrigna Search Engine in Department of Computer Science 2013, Adis Ababa

[40]. Agustina Situmorang and Tima Mariany Arifin, Derivational and Inflectional Morphemes in Pak-Pak Language

[41]. Martin H. and Andrea D. Sims, Understanding Morphology. 2nd edition ed. 2002, London Oxford University Press.

[42]. Getahun A, the Analysis of Ambiguity in Amharic. JES, 2001. Vol XXXIV (2).

[43]. Brown, Peter F., Pietra, Stephen A. Della, Pietra, Vincent J. Della, and L., Robert. Word-sense disambiguation using statistical methods. in In Proceedings of the29th Annual Meeting of the ACL. 1991.

[44]. Yonnas F., Development of stemming algorithm for tigrigna text, in Department of Information Science, Addis Ababa, Ethiopia, 2011.

[45]. Wahiba Ben A, A New Stemmer to Improve Information Retrieval. International Journal of Network Security & Its Applications (IJNSA), 2013. Vol-5.

[46]. Yacob, D., System for Ethiopic Representation in ASCII (SERA). 1996 [cited 2016 Accessed on 3 may]; Available from: http://www.abyssiniacybergateway.net/fidel/

[47]. Zhao Y. and Karypis G., Empirical and theoretical comparisons of selected criterion functions for document clustering. Machine Learning, 2004. Vol 55(3): p. 311-331.

[48]. Forster R., "Document clustering in large German corpora using natural language processing", Ph.D. dissertation, 2006, University of Zurich.

[49]. Manning C. et'al, "Introduction to Information Retrieval", 2008: Cambridge University Press

[50]. Witten and Frank, Data Mining: Practical Machine Learning Tools and Techniques. Second: ed2005: Morgan Kaufmann publications.

[51]. Marine, C., Dekai, W.U.,(2005) "Word Sense Disambiguation vs. Statistical Machine Translation", Proceedings of the 43rd Annual Meeting of the ACL ,Ann Arbor, June 2005

[52]. Miller, G. A., "WordNet: An On-line Lexical Database," Communications of the ACM, Vol.38 No. 11, 1995

[53]. Andres M. et'al "Combining Knowledge- and Corpus-based Word-Sense-Disambiguation Methods", Journal of Artificial Intelligence Research 23 (2005) 299-330, Dept. of Software and Computing Systems, University of Alicante, Spain, published 03/05

[54]. Michael H. "A Genetic Algorithm Using Semantic Relations for Word Sense Disambiguation", University of Colorado, 2011

[55]. Abhishek F. and, Dr.ManojB C. "An Approach for Word Sense Disambiguation using modified Naïve Bayes Classifier", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 4, Nagpur , India, April 2014

[56]. Kolte, S. and, G.Bhirud, S. (2008) "Word Sense Disambiguation Using WordNet Domains", First International Conference on Digital Object Identifier, pp. 1187-1191.

[57]. Jain, "Data clustering: a review. ACM Computing Surveys", 1999. Vol 31(3): p. 264–323.

[58]. Han, J. and Kamber, M. , "Data Mining – Concepts and Techniques". 2001: Morgan K.

[59]. Kaufmann, L. and Rousseeuw, P. J., Clustering by means of medoids, in In Dodge, Y. (Ed.) Statistical Data Analysis based on the L 1 Norm.1987: Elsevier/North Holland, Amsterdam. p. 405–416.

[60]. Dempster, A., laird, N., and Rubin, D., Maximum likelihood from incomplete data via the EM algorithm. . Roy. Statist. Soc. , 1977. Vol 39: p. 1-38.

[61]. Guha S. *et'al* "A robust clustering algorithm for categorical attributes. In Proceedings of ICDE". pp. 512–521. 1999. Sydney, Australia.

[62]. Anjali M. and Babu A. " Ambiguities in Natural Language Processing" Vol.2, Special Issue 5, Kannur University, Kerala, India, October 2014.

APPENDIXES

Section 1

Appendix 1 countries with significant Tigrigna language speakers

Total population (10,139,400)	
Regions with significant population	
Ethiopia	6,316,500
Eritrea	3,430,00,
Italy	54,000
Sudan	43,000
Germany	26,000
Israel	20,000
United States	20,000
Yemen	9,900
Canada	9,300

Appendix 2 List of dialect based Tigrigna stop words

ኣብዚ/ ኣውዚ	እንተኾነግን	ምኳኑን	ከምዘሎ/	ከምዘየሎ/
ኣይቲ	ኮይኑ/ ኾይኑ	እዋን	ከምዘየለ	
ኮይኖም	ብዝተረፈ	ወ/ሮ	ካብ/ ኻብ	
ይኹን	እንተዘይኮይኑ/	ኣይተ	ሓንቲ	
ከምኡ'ውን	እንተዘይኾይኑ	ወ/ት	ይዕበ	
ከይኮነ/ ከይኾነ	ስለዚ	ኣቶ	ይንአስ	
ከምኡ	ምእንትዚ	ብሓፈሽኡ/	ዝኸኖ	
እንተዘየሎ/ እንተኣየለ	ስለዚ'ውን	ብኣጠቓለልኡ	ልዑል	
	ምኳኑ		ትሑት	

ከሎ/ ኸሎ /ትልንሆ

ከኸውን/ ኽኸውን/ ልኾን

ኻልኣይ/ ካልኣይ

ኣዝዮም

ሓደ

ናበይ/ ላበይ

ክዓቢ

ክንእስ

ኣለዎ/ እንሀዎ

ክብል/ ክፀውይ

ከለኹ/ ኸለኹ/ እንከለኹ/ ትነሀኹ

ካልኦት

ኣለና/ እንሀና

ናትካ/ ናኣትኻ

ወዲ

ሰበይቲ

ሰብኣይ

ንኣይ/ ንዓይ/ ልኣነ

ዝብል/ ልብል

ኣቱም/ ኣንቱም

ኢልኩም/ ብልኹም

ኢልክን/ ብልኽን

ይኽእል/ ይከአል

ወላ/ ዋላ

ንቐፃሊ

ኮይኑኒ/ ኾኑኒ

ብዙሕ/ መዓት

ጀሚሩ/ ሞኪሩ

ዘለኻ/ ለለኻ/ ልንሀኻ

ዘለኺ/ ለለኺ/ ልንሀኺ

ንሶም/ እቶሙ

ንሳተን/ እተኑ

ኣለካ/ ኣለኻ/ እነሀኻ

ኣለኺ/ እነሀኺ

ንኽምዚ

ኻምዚ/ ካምዚ/ ሃምዚ

ኢለ/ ብለ

ይኸውን/ ይኾን

ምክንያቱ/ ምኽንያቱ

ነዛ

ነቱ

ነይሩ

ፀኒሑ/ ፀንሑ

ኮነ/ ኾነ

እንዳ/ ዳ

ካልእ/ ኻልእ

ዝኾነት/ ልኾነት

ኢና

ምስ

ምስምስ

ኮታስ

ነናይ

ነይርዎም

እተን/ እተኑ

ዳርጋ

ናይዞም/ ናተሙ

ብድሕሪኡ

እዘን/ እዘኑ

ብዘየካ/ ብዘይካ

ብኹላ

ካብዚ/ ኻብዚ

ከምቲ/ ከምቱ

ከምተን

ምስ'ቲ

በጀካ

በጃኻ

ኣዝዩ

ኣዐርዩ

እስከ/ እስኪ

ዳኣ

በቃ

ከለኺ

እንተለኺ

ከለኻ/ እንተለኻ

እንተሎ/ እንከሎ

ንኡስ

ንኣስታት/ ንከባቢ

ዝብል/ ልብል

ዘለዎ/ ለለዎ

ኣስታት/ ከባቢ

ንከኑ/ ንኽኾኑ

ልዕሊ

ትሕቲ

ከምዘሎን/ ከምለሎ

ዘለዋ/ ለለዋ

ሐዚግን/ ሕጂግን/ ሕዩግን/ ሕዚግን

ዛጊድ/ ዘኽቲ

እንድሕር

ከምዘስዕብ

እዘን

ዘለወን/ ለለወን

ብምኺን

እንትኾና

ኮይኖም/ ኾይኖም

ዝረኽቡ/ ልረኽቡ

ኾይኖም/ ኮይኖም

እንተኾነ

ካብኣቶም

ካብዚኣቶም

ካብዚኣተን

ካብኣተን

ሕድሕድኦም

ዝግባእ

ዝኾና

እንዳ

ምስኣም

ሒዞም

ዝኣኽሉ

ዝኣኽላ

ካብዞም

ካብዘን

ሕድሕድ

ይኹኑ

ይኹና

ይኹን

ኣብቲ / ኣፍቲ

ማለት

እዩ /ይኡ

ከም

ኑቲ

ኣብ

ከማኻ

ከምዘለኺ

ከምዘለዎ

እቲ / እትው

ወይ

ድማ / ደማ

ይኽእሉ

ይኽእላ

ካልእ/ ኻልእ/ ካልእ

ከምቲ/ ካምቲ

በተን

ብብዙሕ

ናይዚኦም

ምኟንኩም

ኢሎም

እንትረኣ

ንምኟን

ቀንዲ/ ዋና

ነዘን/ ንተን

ክሳብ / ክሳዕ/ ክሻብ/ ኽሳብ/ እስካብ

መን

ኣበይ

እምበር

በቶም

ነዚ

ይኹን

ኣሎ /እነሆ

እዚ

ዝኾነ /ልኾነ

ኸዓ /ክኣ/ ከዓ/ ኽኣ/ ለለ

በዞም

ኣለውዎ

እንተ

ስለዘለውና

ብተወሳኺ

ከምዝነበረ

ዘይኮነስ

ካብዚኦም

ሳላ/ ሻላ/ ብሳላ

ትኹን

ዛዕባና

ኣብቶም

ኣብተን

በብ/ በቢ

እዛ /እዛው

ኣይኮነን

እንታይ / ታይ

እታ

ካብ

ነይሮም / ፀንሖም

ከምቶም

እቶም/ እቶሙ

ከምዘለኒ

ከምዘለኩም

እውን

ውን

ምስተን/ ምስቶም

ግደ/ እጃም

መላእ

ኮይንካ

እምብኣር

ብእኡ/ ብኡ

ከምዝስዕብ

ብምሉእ

ኾይኖም/ ኮይኖም

ይህልዎም

ኣይኮነትን

ዳርጋ

ከቶ

ኣብዚ / ኣብዝው/ ኣብታው

ነበሩ

ነበረ

ኩሉ/ ብኹሉ/ ኹሉ

ኣነ

ንስኻ

ንስኺ

ዘለው /ለለው

ንሱ

ንሳ

ንሕና

ንስኻትኩም

ንስኻትክን

ስጋዕ

ብዛዕባ

ከለዋ

ነታ

ናብ

ዘለኒ

ግን

ጥራሕ

ካብቶም

እኳ /ኳ

ምኽኖም

ብኡኣም

በቲ

ብቲ

ክኾና

ብዘይ /ብለይ

ምእንቲ

ስለ

እሞ

ዘለኩም

ድሕሪ

ቅድሚ

በዚ

ብዚ

በኡኣም

በዙይ

ናይዞም

ኩሎም

ኢሎም

ዘለክን

ናይቲ

እንተድኣ

ናይ

ዘለዎም / ለለዎም

ከምኡውን

እዮም

ካብተን

ድኣ

ናይዞም

ናይዘን

ናይቶም

ናይተን

ኣብዚኣም

እዙይ/ እዝው/

እዚኣ/ እዝዋ

ዘለዋ

የብለንን

የብለይን

እንተዝኾና

እዚኣም

ዘለወን

ከምዘለኪ

ስለዝኾነ / ስለዝኮነ

ነይርወን

ዘለወን

ዝባሃላ

ምስቲ

ዘለኪ

ምኽና

የለን/ የልቦን/ የለይ

ኮይኑ

ስለዝኾነውን

ከምኡ /ኸምኡ /ሃምኡ

ምኽን

የለዋን

ምስ /ምስምስ

ከማይ

የብሉን

ከማና

ኩለን

እያ /ይኣ

ናይዘን

ናይቶም

ዝኾነኮይኑ

ኣብዛ

ኣብዚኣ

ኣላ /እንሃ

የብልናን

የብልካን

ከምተን

ኢሉ

ኢላ

ኢለን

ኣይኮኑን

የብሎምን

ከምዚኣቶም

ከምዚኣተን

ኣይኮነትን

ከምቶም

ዘለና

ዘለካ

ዘሎ / ልንሆ/ ድንሆ

የለዉን

ዘላ

ይኹን እምበር

ይኹን ደኣ ምበር

የብላን

ወዘተ/ ካልኣትእውን

ስለዝኾኑ

ስለዝኾና

ስለዝኾነት

ስለዝኾንኩ

ብምኽንኩም

ቅድሚ

ቅድሚት

ከምዘለና

ከምዘለኻ

ከምዘለኽን

ናትና /ናኣትና

ናታትክን

ንሳተን

ነቱይ

ከምአን

እየን

ኩላተን

ከምዘለኹም

ከምዙይ / ከምዚ/ ሃምዚ/ ኸምዚ

ከምዚኣ

ብቅድሚት

ብድሕሪት

ብቅድሚ

ድሕሪ

ንቶም

ንተን

ናታትኩም

ናታቶም

ናታተን

ማለተን

ብምኺና

ዘይኮነ

ንኸይህልዎም

ዝኣመሳሰሉ/ ዝመሰሉ/ ልበሉ

ብምኺንክን

ብምኺንኪ

ብምኺኑ

ነቶም / ነዞም

ከከም

ኩሉኹም

ኩልኽን

ስለዝኾንና

ናይዚ

ዝበሃሉ

ናታ

ናቱ

ናይቱ

መዓዝ

ምስዚ

ምስኡ

የብልኩምን

የብልክንን

ከለና

ክብሉ

ኩልኻትክን

ከማኺ

ብምኺንና

ኣቢሉ

ኣቢላ

ዛዕባ

እስኪ

ስለዝኾንክን

ከለው

ከምዘለካ

ስለዝኾንኩም

ኩላትና

ከማኽን

ከማኹም

ከምኣቶም

ከምዘለዋ

ምስኣ

ምሳና

ብምኺነን

ምሳኹም

ምሳኽን

ማለትኩም

እንትኸውን

እንትኾን

እንተዝኸውን

ማለትክን

ምስኦም

ማለቶም

ማለትና

ማለትኪ

ብኸመይን

ከይኮነ/ ተይኮነ

ብዙሓት

ውሑዳት

ከምኣተን

ኣብዝሓ

ብኣብዝሓ

ካሊእ

ክንዲቲ

ከምታ

ካልኦት

ከምዘለክን

ማለትካ

ምስኣቶም

ምስኣተን

ብምኺኖም

ማለታ

ማለቱ

ንማለት

ምስኡ

ኩሉኻትኩም

እማ

እንተዝኾኑ

እንተዝኾን

ብዝኾ

ከሰዐይ

ዝሓዘ/ ልሓዘ

ዋንኡ

ኣይኾንን

ኣይከነን

ይኹንዳአምበር ኢኹም ዶ ኣብዚኣማ/ እዛው

ጭራሽ

እንተሃለወ/ እንተሃለየ ንዕኡ ምስከደ

ምኻነን ኣይኮንኩን

ካማን ብሓፈሻ

Appendix 3 List of dialect based Tigrigna Prefixes

ም	ነን	ዘይተ /ለይተ	እንተይ
ብ	ንም	ከምት	ከምእን
ን	ምስ	ኣይተ	ከምዘይ/ ከምለይ
ና	ንዝ	ኣይን	እንተዝ/ እንተል
ዝ / ል/ ዚ	ብዝ /ብል	ኣይተ	እንተዘይ /እንተለይ
ኣ	ዘይ /ለይ	ኣይት	ከምዘይተ /ከምለይተ
ኢ	እን	ከይተ	እንተይተ
ከ / ኪ/ ኽ	ክት	ከምዝ / ከምል	እንድሕር
ተ	እና	እናተ / ናተ	እንተዘይተ /እንተለይተ
ይ	ክን	እንት	እንድሕርዘይ
ብም	ተተ	እንተ / እተ	/እንድሕርለይ
በቢ	ስለ	እንዳ	እንድሕርዘይተ
ከየ	ከም	ስለዘይ / ስለለይ	/እንድሕርለይተ
ኣይ	ዝተ /ልተ	ስለዝ/ ስለል	

Appendix 4 List of dialect based Tigrigna Suffixes

ና	ኡ	ዎ	ነን
ት	ኛ	ቲ	ኩም
ን	ኻ	ታት	ኹም
ተ	ኺ	ኣን	ናን
ሉ	ም	ያዊ	ክን
ዊ	ቶ	ውቲ	ዊን

ነት	ቶም	ያዊን	ናለን
ኣን	ኣም	ናዮም	ውነቱ
ውን	ሎም	ትለይ	ላዋን
ለይ	ኩምን	ታቶም	ያዊያን
ለን	ውቲና	ናለይ	ዊያን
ተኛ	ውቶም	ነታዊ	ያውያን
ትን	ውተን	ታትን	ውያን
ምን	ትኣም	ሎምን	ውትኹም

Appendix 5 List of dialect based Tigrigna Circumfixes

ኣይ - ን	ኣይ - ኩን	ኣይም - ኽንን	ክን - ሎም
የ - ን	ኣይ - ካን	ኣይ - ለይን	ከምዝ - ዎ
ኣይ - ትን	ኣይ - ክን	ንዝ - ሉ	ክት - ሉ
የ - ትይ	ኣይት - ን	ዝ - ን	ከምዘይ - ት
ይ - ሎም	ኣይም - ኹን	እት - ሉ	ከምዘይ - ን
ኣይ - ናን	ኣይም - ኩን	እት - ን	
የ ትን	የ ክን	የ ለይን	ከምለይ ት
ለይ ት	የት ን	ልል ሉ	ኣይ እኹን
ኣይ ትይ	የም ኹን	ል ን	የ እኹን
የ ናን	የም ኩን	ት ሉ	መ ቲ
የ ኩን	የም ኽንን	ከምል ዎ	
የ ካን	የም ኹምን	ክት ሎም	

Appendix 6 List of collected Tigrigna Infixes

ዳ ራ ፉ

ባ ጋ ል

ታ ፃ ማ

ዋ ጣ

Appendix 7 Tigrigna Short words

ቤት ት/ቲ = ቤት ትምህርቲ

ቤት ፍ/ዲ = ቤት ፍርዲ

ት/ቲ = ትምህርቲ

ክፍለ ት/ቲ = ክፍለ ትምህርቲ

ሃ/ስላሴ = ሃይለስላሴ

መ/ር = መምህር

ወ/ር = ወታደር

ወ/ሮ = ወይዘሮ

ወ/ሪት = ወይዘሪት

ወ/ስላሴ = ወልደስላሴ

ፍ/ስላሴ = ፍቅረስላሴ

ቤት ፅ.ት = ቤት ፅሕፈት

ፕ/ር = ፕሮፌሰር

ቀ.ሚንስትር = ቀዳማይ ሚኒስትር

ዶ/ር = ዶክተር

ገ/ጊዮርጊስ = ገብረጊዮርጊስ

ቤ/ክርስትያን = ቤተ ክርስትያን

ም/ኣቦወንበር = ምክትል ኣቦወንበር

ቤት ም/ሪ = ቤት ምኽሪ

ተ/ሃይማኖት = ተክለሃይማኖት

ሚ/ር = ሚኒስቴር

ኮ/ል = ኮሌኔል

ሜ/ጄነራል = ሜጀር ጄነራል

ብ/ጄነራል = ብርጋዴር ጄነራል

ሌ/ኮለኔል = ሌቴናል ኮለኔል

ኣ/ኣ = ኣዲስ ኣበባ

ሓ/ማሕበር = ሓረስቶት ማሕበር

ደ.ኣንስትዮ =ደቂ ኣንስትዮ

ኢ/ያ = ኢትዮድያ

ገ/ልምዓት = ገጠር ልምዓት

ሕ.ወኪል = ሕርሻ ወኪል

ላ/ማይጨው = ላዕላይ ማይጨው

ታ.ማይጨው = ታሕታይ ማይጨው

ገ/ማርያም = ገብረማረያም

ገ/ዚሄር = ገረዚሄር

ሓ/ዓሰርተ = ሓለቓ ዓሰርተ

ሓ.ሚኢቲ = ሓለቓ ሚኢቲ

ሓ.ሽሕ = ሓለቓ ሽሕ

ሓ.ዘመን = ሓዲሽ ዘመን

ር/ምምሕዳር = ርእሰ ምምሕዳር

ዓ/ግ = ዓድግራት

ዕ.ሓሙስ = ዕዳጋ ሓሙስ

ማ/ጨው = ማይጨው

ማ/ሰብ = ማሕበረ ሰብ

ዓ.ዓ = ዓመተ ዓለም

ማ/ኮሚቴ = ማእኸላይ ኮሚቴ

ር/መምህር = ርእሰ መምህር

ፕ/ት = ፕሬዚዳንት

ሃ.ተፈጥሮ = ሃፍቲ ተፈጥሮ

ቤት ፍ/ሒ = ቤት ፍትሒ

ሚ/ሕርሻ = ሚኒስቴር ሕርሻ

ቤት ህ/ት = ቤት ህንፀት

ር/ከተማ = ርእሰ ከተማ

ዓ.ም = ዓመተ ምህረት

Appendix 8 Ethiopic Unicode Representation [1]

Appendix 9 The Tigrigna alphabet ('Fidel') [16]

	Ordinary characters							Diphthong (‘diqala’) characters				
1	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
	he	hu	hi	ha	hE	h	ho					
2	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ			ሏ		
	le	lu	li	la	lE	l	lo			lWa		
3	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ			ሗ		
	He	Hu	Hi	Ha	HE	H	Ho			HWa		
4	መ	ሙ	ሚ	ማ	ሜ	ም	ሞ			ሟ		
	me	mu	mi	ma	mE	m	mo			mWa		
5	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ			ሧ		
	`se	`su	`si	`sa	`sE	`s	`so			`sWa		
6	ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ			ሯ		
	re	ru	ri	ra	rE	r	ro			rWa		
7	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ			ሷ		
	se	su	si	sa	sE	s	so			sWa		
8	ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ			ሿ		
	xe	xu	xi	xa	xE	x	xo			xWa		
9	ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቍ	ቋ	ቌ	ቊ
	qe	qu	qi	qa	qE	q	qo	qWe	qWu	qWa	qWE	qWi
10	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ			ቧ		
	be	bu	bi	ba	bE	b	bo			bWa		
11	ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ			ቯ		
	ve	vu	vi	va	vE	v	vo			vWa		
12	ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ			ቷ		

	te	tu	ti	ta	tE	t	to			tWa		
13	ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ			ቿ		
	ce	cu	ci	ca	cE	c	co			cWa		
14	ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ኊ	ኋ	ኌ	ኍ
	`he	`hu	`hi	`ha	`hE	`h	`ho	hWe	hWu	hWa	hWE	hWi
15	ነ	ኑ	ኒ	ና	ኔ	ን	ኖ			ኗ		
	ne	nu	ni	na	nE	n	no			nWa		
16	ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ			ኟ		
	Ne	Nu	Ni	Na	NE	N	No			NWa		
17	አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ			ኧ		
	e	u	i	a	E	I	o			ea		
18	ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኰ	ኩ	ኳ	ኲ	ኵ
	ke	ku	ki	ka	kE	k	ko	kWe	kWi	kWa	kWE	kWu
19	ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ					
	`ke	`ku	`ki	`ka	`kE	`k	`ko					
20	ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ					
	we	wu	wi	wa	wE	w	wo					
21	ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ					
	`e	`u	`i	`a	`E	`I	`o					
22	ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ			ዟ		
	ze	zu	zi	za	zE	z	zo			zWa		
23	ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ			ዧ		
	Ze	Zu	Zi	Za	ZE	Z	Zo			ZWa		
24	የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ					
	ye	yu	yi	ya	yE	y	yo					

25	ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ			ዷ		
	de	du	di	da	dE	d	do			dWa		
26	ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ			ጇ		
	je	ju	ji	ja	jE	j	jo			jWa		
27	ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጉ	ጕ	ጓ	ጔ
	ge	gu	gi	ga	gE	g	go	gWe	gWu	gWi	gWa	gWE
28	ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ				ጧ	
	Te	Tu	Ti	Ta	TE	T	To				TWa	
29	ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ				ፏ	
	Ce	Cu	Ci	Ca	CE	C	Co				CWa	
30	ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ					
	Pe	Pu	Pi	Pa	PE	P	Po					
31	ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ				ጿ	
	Se	Su	Si	Sa	SE	S	So				SWa	
32	ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
	`Se	`Su	`Si	`Sa	`SE	`S	`So					
33	ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ				ፏ	
	fe	fu	fi	fa	fE	f	fo				fWa	
34	ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ				ፗ	
	pe	pu	pi	pa	pE	p	po				pWa	

Appendix 10 Selected ambiguous words and their Tigrigna meaning

ተ.ቁ	ተመሳሳሊ ቃል	ትርጉም
1	መደብ መደብ መደብ	እቅድ ባህላዊ መደቀሲ ኣብ ሓደ ምጉጃል
2	ሓለፈ ሓለፈ ሓለፈ ሓለፈ	ሓሊቃ ተዘዋዊሩ(ካብ ክላስ ናብ ክላስ) ሞተ ከደ
3	ከበረ ከበረ	ዋጋ ወሰከ ክብሪ ተዉሃቦ
4	ተመነየ ተመነየ	ተስፋ ገበረ ሰልቸየ፣ኣፅለአ
5	መተረ መተረ	ዓቀነ ከተፈ
6	ዓረቐ ዓረቐ	ኣስማመዐ ተቸገረ
7	ሃደመ ሃደመ	ጎየየ ገዛ ሰረሐ(ህድሞ)

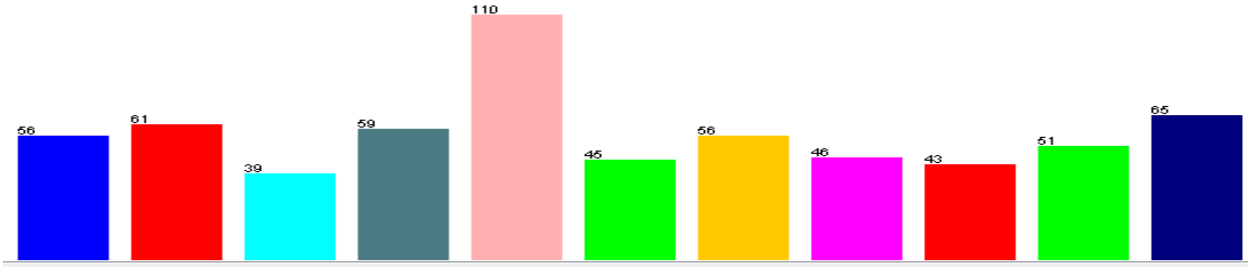
Section 2

Appendex 11. The effect of stemming screen shoot input (20009) output (14289)

Appendex 12. Screen shoot Bargraph of the different classes from weka 3.8.1

No.	Label	Count	Weight
1	program	56	56.0
2	traditional_bed	61	61.0
3	grouping	39	39.0
4	promote	59	59.0
5	pass	110	110.0
6	boss	45	45.0
7	pass_away	56	56.0
8	running	46	46.0
9	building_house	43	43.0
10	expensive	51	51.0
11	respecting	65	65.0

Class: class (Nom) Visualize All



Appendex 12 saple of converted to .arff screen shoots of the corpus

arff screen shoots of the corpus 1 – 1 and 2 -2 of words to the left side of target word and right side of the target word

File Edit View

1-1.arff *

Relation: book

No.	1: left1 String	2: target String	3: right1 String	4: class Nominal
44	ake...	medeb	tmhrty	program
45	mer...	medeb	Hizom	program
46	key...	medeb	tezey...	program
47	abe...	nme...	etot	program
48	kifte...	med...	limeat	program
49		medeb	limeat	program
50	kal...	med...	tetse...	program
51	ke...	med...	emo	program
52		med...	Eted...	program
53	tigray	med...	ming...	program
54	kire...	medeb	qulfiy	program
55	tmh...	med...	nay	program
56	zha...	medeb	teqe...	program
57	nea...	teba...	ab	traditional_bed
58	kifly	medeb	zneb...	traditional_bed
59	zwuel	medeb	kekem	traditional_bed
60	aw...	medeb	kekem	traditional_bed
61	egri	med...	kof	traditional_bed
62	zke...	medeb	iyu	traditional_bed
63	zse...	medeb	keyte...	traditional_bed
64	zke...	medeb	teten...	traditional_bed
65	me...	medeb	alo	traditional_bed
66	nEs...	medeb	kisra...	traditional_bed
67	zke...	medeb	tezey...	traditional_bed
68	chiqa	medeb	kihlyo	traditional_bed
69	me...	medeb	zelewo	traditional_bed

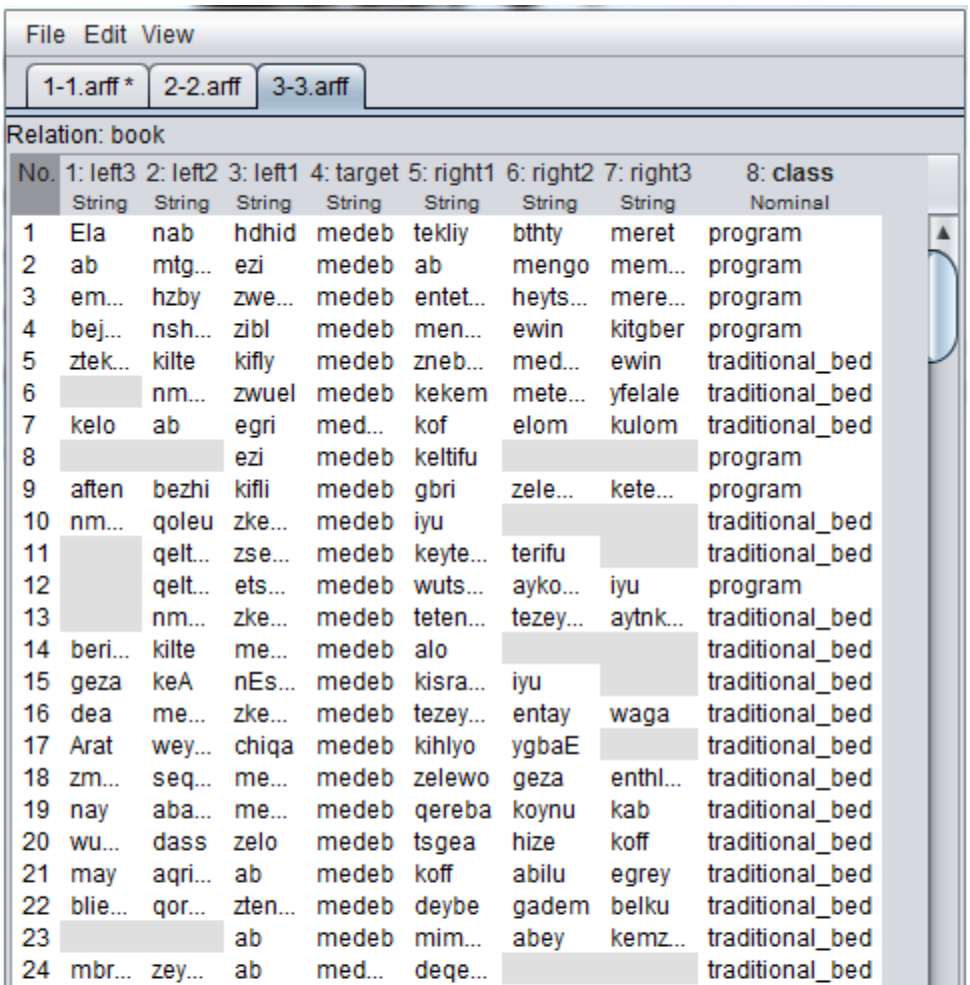
File Edit View

1-1.arff * 2-2.arff

Relation: book

No.	1: left2 String	2: left1 String	3: target String	4: right1 String	5: right2 String	6: class Nominal
1	nab	hdhid	medeb	tekliy	bthty	program
2	mtg...	ezi	medeb	ab	mengo	program
3	hzby	zwe...	medeb	entet...	heyts...	program
4	nsh...	zibl	medeb	men...	ewin	program
5	kilte	kifly	medeb	zneb...	med...	traditional_bed
6	nm...	zwuel	medeb	kekem	mete...	traditional_bed
7	ab	egri	med...	kof	elom	traditional_bed
8		ezi	medeb	keltifu		program
9	bezhi	kifli	medeb	gbri	zele...	program
10	qoleu	zke...	medeb	iyu		traditional_bed
11	qelt...	zse...	medeb	keyte...	terifu	traditional_bed
12	qelt...	ets...	medeb	wuts...	ayko...	program
13	nm...	zke...	medeb	teten...	tezey...	traditional_bed
14	kilte	me...	medeb	alo		traditional_bed
15	keA	nEs...	medeb	kisra...	iyu	traditional_bed
16	me...	zke...	medeb	tezey...	entay	traditional_bed
17	wey...	chiqa	medeb	kihlyo	ygbaE	traditional_bed
18	seq...	me...	medeb	zelewo	geza	traditional_bed
19	aba...	me...	medeb	qereba	koynu	traditional_bed
20	dass	zelo	medeb	tsgea	hize	traditional_bed
21	aqri...	ab	medeb	koff	abilu	traditional_bed
22	qor...	zten...	medeb	deybe	gadem	traditional_bed
23		ab	medeb	mim...	abey	traditional_bed
24	zey...	ab	med...	deqe...		traditional_bed
25	ms...	kab	med...	tesiom	msm...	traditional_bed
26	kul...	me...	med...	4	meaz...	traditional_bed

arff screen shoots of the corpus 3 - 3 of words to the left side of target word and right side of the target word



File Edit View

1-1.arff * | 2-2.arff | 3-3.arff

Relation: book

No.	1: left3 String	2: left2 String	3: left1 String	4: target String	5: right1 String	6: right2 String	7: right3 String	8: class Nominal
1	Ela	nab	hdhid	medeb	tekliy	bthty	meret	program
2	ab	mtg...	ezi	medeb	ab	mengo	mem...	program
3	em...	hzby	zwe...	medeb	entet...	heyts...	mere...	program
4	bej...	nsh...	zibl	medeb	men...	ewin	kitgber	program
5	ztek...	kilte	kifly	medeb	zneb...	med...	ewin	traditional_bed
6		nm...	zwuel	medeb	kekem	mete...	yfelale	traditional_bed
7	kelo	ab	egri	med...	kof	elom	kulom	traditional_bed
8			ezi	medeb	keltifu			program
9	aften	bezhi	kifli	medeb	gbri	zele...	kete...	program
10	nm...	qoleu	zke...	medeb	iyu			traditional_bed
11		qelt...	zse...	medeb	keyte...	terifu		traditional_bed
12		qelt...	ets...	medeb	wuts...	ayko...	iyu	program
13		nm...	zke...	medeb	teten...	tezey...	aytnk...	traditional_bed
14	beri...	kilte	me...	medeb	alo			traditional_bed
15	geza	keA	nEs...	medeb	kisra...	iyu		traditional_bed
16	dea	me...	zke...	medeb	tezey...	entay	waga	traditional_bed
17	Arat	wey...	chiqa	medeb	kihlyo	ygbaE		traditional_bed
18	zm...	seq...	me...	medeb	zelewo	geza	enthl...	traditional_bed
19	nay	aba...	me...	medeb	qereba	koynu	kab	traditional_bed
20	wu...	dass	zelo	medeb	tsgea	hize	koff	traditional_bed
21	may	aqri...	ab	medeb	koff	abilu	egrey	traditional_bed
22	blie...	qor...	zten...	medeb	deybe	gadem	belku	traditional_bed
23			ab	medeb	mim...	abey	kemz...	traditional_bed
24	mbr...	zey...	ab	med...	deqe...			traditional_bed

arff screen shoots of the corpus 4 - 4 of words to the left side of target word and right side of the target word

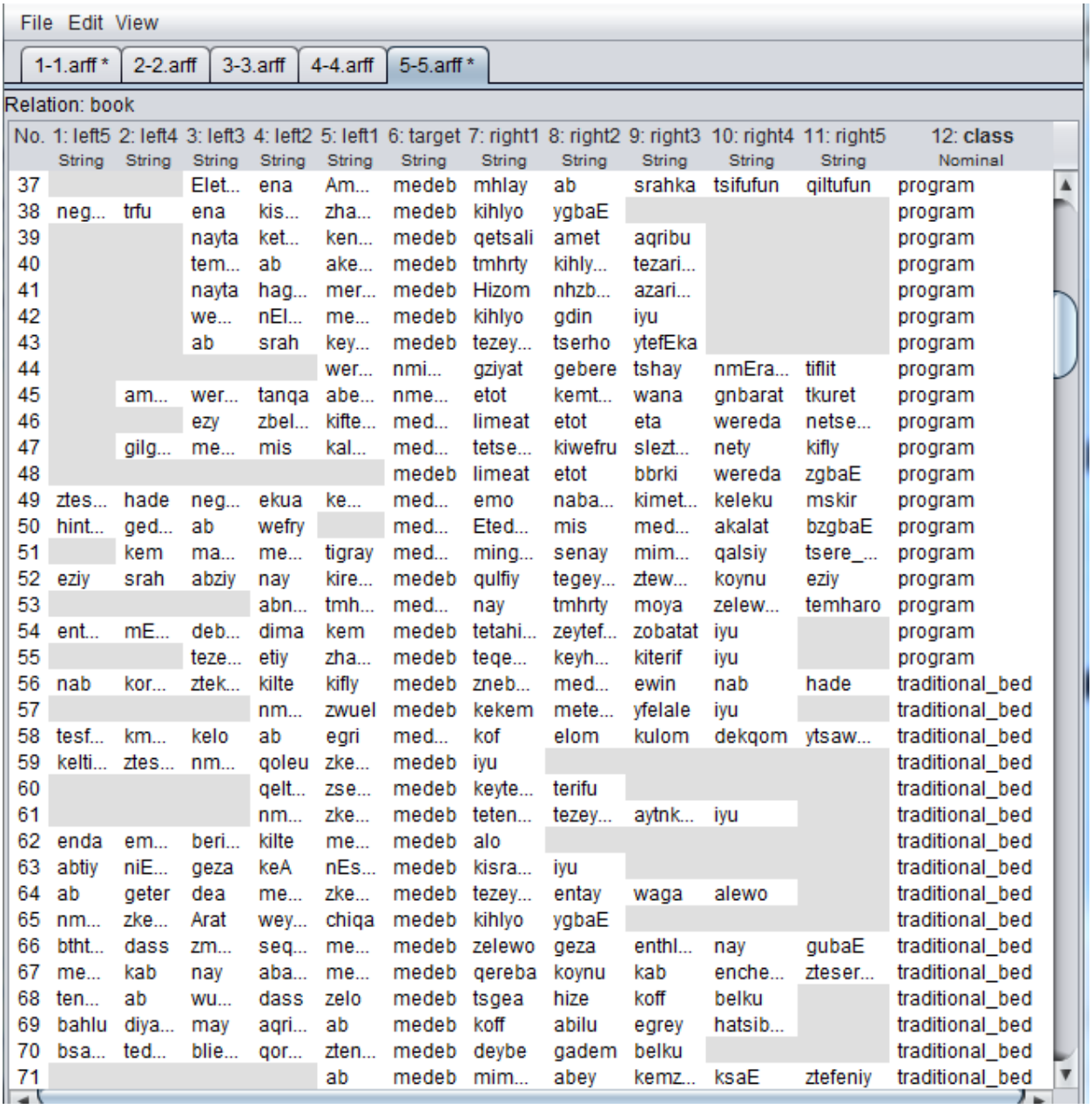
File Edit View

1-1.arff * | 2-2.arff | 3-3.arff | 4-4.arff

Relation: book

No.	1: left4 String	2: left3 String	3: left2 String	4: left1 String	5: target String	6: right1 String	7: right2 String	8: right3 String	9: right4 String	10: class Nominal
1	kab	Ela	nab	hdhid	medeb	tekliy	bthty	meret	may	program
2	gna	ab	mtg...	ezi	medeb	ab	mengo	mem...	rEsa...	program
3	ykun	em...	hzby	zwe...	medeb	entet...	heyts...	mere...	yelen	program
4	ata...	bej...	nsh...	zibl	medeb	men...	ewin	kitgber	zkEl	program
5	kor...	ztek...	kilte	kifly	medeb	zneb...	med...	ewin	nab	traditional_bed
6			nm...	zwuel	medeb	kekem	mete...	yfelale	iyu	traditional_bed
7	km...	kelo	ab	egri	med...	kof	elom	kulom	dekq...	traditional_bed
8				ezi	medeb	keltifu				program
9	zelo	aften	bezhi	kifli	medeb	gbri	zele...	kete...	iyu	program
10	ztes...	nm...	qoleu	zke...	medeb	iyu				traditional_bed
11			qelt...	zse...	medeb	keyte...	terifu			traditional_bed
12			qelt...	ets...	medeb	wuts...	ayko...	iyu		program
13			nm...	zke...	medeb	teten...	tezey...	aytnk...	iyu	traditional_bed
14	em...	beri...	kilte	me...	medeb	alo				traditional_bed
15	niE...	geza	keA	nEs...	medeb	kisra...	iyu			traditional_bed
16	geter	dea	me...	zke...	medeb	tezey...	entay	waga	alewo	traditional_bed
17	zke...	Arat	wey...	chiqa	medeb	kihlyo	ygbaE			traditional_bed
18	dass	zm...	seq...	me...	medeb	zelewo	geza	enthl...	nay	traditional_bed
19	kab	nay	aba...	me...	medeb	qereba	koynu	kab	ench...	traditional_bed
20	ab	wu...	dass	zelo	medeb	tsgea	hize	koff	belku	traditional_bed
21	diya...	may	aqri...	ab	medeb	koff	abilu	egrey	hatsi...	traditional_bed
22	ted...	blie...	qor...	zten...	medeb	deybe	gadem	belku		traditional_bed
23				ab	medeb	mim...	abey	kemz...	ksaE	traditional_bed
24	hil...	mbr...	zey...	ab	med...	deqe...				traditional_bed
25	zne...	zer...	ms...	kab	med...	tesiom	msm...	fege...	nabay	traditional_bed
26			kul...	me...	med...	4	meaz...	koynu	teme...	traditional_bed
27	esty	nea	me...	zke...	medeb	kerEy...	beluny			traditional_bed
28	aba	me...	alEl...	zdqi...	medeb	msAr...	nabty	meEr...	gasha	traditional_bed
29	eziy	elom	nab	zea...	med...	keyd...	kef	belu		traditional_bed
30	ms	geb...	nab	lieleu	med...	gadem	belku			traditional_bed
31	tselot	ena...	ane...	ab	med...	koyne	tesh...	nzrey	dima	traditional_bed
32	nskin	dah...	had...	ele	med...	hadige	tetew	ele	kidan...	traditional_bed
33	sha...	hize	kab	zdq...	med...	telaele	nmkad	wese...		traditional_bed
34	adey	teze...	nlo...	abzia	medeb	tezeh...	zdele...	hisab	kikefil	traditional_bed
35	abeu	zne...	sebat	dima	nme...	tay	chigr	alewo	elom	traditional_bed

arff screen shoots of the corpus 5 – 5 of words to the left side of target word and right side of the target word



File Edit View

1-1.arff * | 2-2.arff | 3-3.arff | 4-4.arff | 5-5.arff *

Relation: book

No.	1: left5 String	2: left4 String	3: left3 String	4: left2 String	5: left1 String	6: target String	7: right1 String	8: right2 String	9: right3 String	10: right4 String	11: right5 String	12: class Nominal
37			Elet...	ena	Am...	medeb	mhlay	ab	srahka	tsifufun	qiltufun	program
38	neg...	trfu	ena	kis...	zha...	medeb	kihlyo	ygbaE				program
39			nayta	ket...	ken...	medeb	qetsali	amet	aqribu			program
40			tem...	ab	ake...	medeb	tmhrty	kihly...	tezari...			program
41			nayta	hag...	mer...	medeb	Hizom	nhzb...	azari...			program
42			we...	nEl...	me...	medeb	kihlyo	gdin	iyu			program
43			ab	srah	key...	medeb	tezey...	tserho	ytefEka			program
44					wer...	nmi...	gziyat	gebere	tshay	nmEra...	tiflit	program
45		am...	wer...	tanqa	abe...	nme...	etot	kemt...	wana	gnbarat	tkuret	program
46			ezy	zbel...	kifte...	med...	limeat	etot	eta	wereda	netse...	program
47		gilg...	me...	mis	kal...	med...	tetse...	kiwefru	slezt...	nety	kifly	program
48						medeb	limeat	etot	bbrki	wereda	zgbaE	program
49	ztes...	hade	neg...	ekua	ke...	med...	emo	naba...	kimet...	keleku	mskir	program
50	hint...	ged...	ab	wefry		med...	Eted...	mis	med...	akalat	bzgbaE	program
51		kem	ma...	me...	tigray	med...	ming...	senay	mim...	qalsiy	tsere_...	program
52	eziy	srah	abziy	nay	kire...	medeb	qulfiy	tegey...	ztew...	koynu	eziy	program
53				abn...	tmh...	med...	nay	tmhrty	moya	zelew...	temharo	program
54	ent...	mE...	deb...	dima	kem	medeb	tetahi...	zeytef...	zobatat	iyu		program
55			teze...	etiy	zha...	medeb	teqe...	keyh...	kiterif	iyu		program
56	nab	kor...	ztek...	kilte	kifly	medeb	zneb...	med...	ewin	nab	hade	traditional_bed
57				nm...	zwuel	medeb	kekem	mete...	yfelale	iyu		traditional_bed
58	tesf...	km...	kelo	ab	egri	med...	kof	elom	kulom	dekqom	ytsaw...	traditional_bed
59	kelti...	ztes...	nm...	qoleu	zke...	medeb	iyu					traditional_bed
60				qelt...	zse...	medeb	keyte...	terifu				traditional_bed
61				nm...	zke...	medeb	teten...	tezey...	aytnk...	iyu		traditional_bed
62	enda	em...	beri...	kilte	me...	medeb	alo					traditional_bed
63	abtiy	niE...	geza	keA	nEs...	medeb	kisra...	iyu				traditional_bed
64	ab	geter	dea	me...	zke...	medeb	tezey...	entay	waga	alewo		traditional_bed
65	nm...	zke...	Arat	wey...	chiqa	medeb	kihlyo	ygbaE				traditional_bed
66	btht...	dass	zm...	seq...	me...	medeb	zelewo	geza	enthl...	nay	gubaE	traditional_bed
67	me...	kab	nay	aba...	me...	medeb	qereba	koynu	kab	enche...	zteser...	traditional_bed
68	ten...	ab	wu...	dass	zelo	medeb	tsgea	hize	koff	belku		traditional_bed
69	bahlu	diya...	may	aqri...	ab	medeb	koff	abilu	egrey	hatsib...		traditional_bed
70	bsa...	ted...	blie...	qor...	zten...	medeb	deybe	gadem	belku			traditional_bed
71					ab	medeb	mim...	abey	kemz...	ksaE	ztefeniy	traditional_bed

arff screen shoots of the corpus 6 - 6 of words to the left side of target word and right side of the target word

File Edit View

1-1.arff * | 2-2.arff | 3-3.arff | 4-4.arff | 5-5.arff * | 6-6.arff

Relation: book

No.	1: left6 String	2: left5 String	3: left4 String	4: left3 String	5: left2 String	6: left1 String	7: target String	8: right1 String	9: right2 String	10: right3 String	11: right4 String	12: right5 String	13: right6 String	14: class Nominal
1	may	key...	kab	Ela	nab	hdhid	medeb	tekliy	bthty	meret	may	zwesid	tbo	program
2		koy...	gna	ab	mtg...	ezi	medeb	ab	mengo	mema...	rEsane	mema...	trah	program
3			ykun	em...	hzby	zwe...	medeb	entet...	heyts...	mered...	yelen			program
4	kab	wu...	ata...	bej...	nsh...	zibl	medeb	men...	ewin	kitgber	zkEl	tsegat...	bzgbaE	program
5	gez...	nab	kor...	ztek...	kilte	kifly	medeb	zneb...	med...	ewin	nab	hade	Aratn	traditional_bed
6					nm...	zwuel	medeb	kekem	mete...	yfelale	iyu			traditional_bed
7	end...	tesf...	km...	kelo	ab	egri	med...	kof	elom	kulom	dekqom	ytsaw...		traditional_bed
8						ezi	medeb	keltifu						program
9	anfet	arEyu	zelo	aften	bezhi	kifli	medeb	gbri	zele...	ketem...	iyu			program
10	ezi	kelti...	ztes...	nm...	qoleu	zke...	medeb	iyu						traditional_bed
11					qelt...	zse...	medeb	keyte...	terifu					traditional_bed
12					qelt...	ets...	medeb	wuts...	ayko...	iyu				program
13					nm...	zke...	medeb	teten...	tezey...	aytnkrin	iyu			traditional_bed
14		enda	em...	beri...	kilte	me...	medeb	alo						traditional_bed
15	me...	abtiy	niE...	geza	keA	nEs...	medeb	kisra...	iyu					traditional_bed
16		ab	geter	dea	me...	zke...	medeb	tezey...	entay	waga	alewo			traditional_bed
17	we...	nm...	zke...	Arat	wey...	chiqa	medeb	kihlyo	ygbaE					traditional_bed
18		btht...	dass	zm...	seq...	me...	medeb	zelewo	geza	enthluw	nay	gubaE	geza	traditional_bed
19	ges...	me...	kab	nay	aba...	me...	medeb	qereba	koynu	kab	enche...	zteser...	geza	traditional_bed
20	sh...	ten...	ab	wu...	dass	zelo	medeb	tsgea	hize	koff	belku			traditional_bed
21	ges...	bahlu	diya...	may	aqri...	ab	medeb	koff	abilu	egrey	hatsib...			traditional_bed
22		bsa...	ted...	blie...	qor...	zten...	medeb	deybe	gadem	belku				traditional_bed
23						ab	medeb	mim...	abey	kemzd...	ksaE	ztefeniy	nguho	traditional_bed
24		enk...	hil...	mbr...	zey...	ab	med...	deqe...						traditional_bed
25	mis...	sebat	zne...	zer...	ms...	kab	med...	tesiom	msm...	fegegt...	nabay	metsu		traditional_bed
26					kul...	me...	med...	4	meaz...	koynu	temes...	qrtsy	alewom	traditional_bed
27			esty	nea	me...	zke...	medeb	kerEy...	beluny					traditional_bed
28	akt...	ent...	aba	me...	alEl...	zdqi...	medeb	msAr...	nabty	meErefy	gasha	aketilo...	wesed...	traditional_bed
29			eziy	elom	nab	zea...	med...	keyd...	kef	belu				traditional_bed
30	tselot	neg...	ms	geb...	nab	lieleu	med...	gadem	belku					traditional_bed
31	tetew	elom	tselot	ena...	ane...	ab	med...	koyne	tesh...	nzrey	dima	moyte	tselot	traditional_bed
32	egzi...	ym...	nskin	dah...	had...	ele	med...	hadige	tetew	ele	kidaney	kielish	reayaniy	traditional_bed
33	ena	2	sha...	hize	kab	zdq...	med...	telaele	nmkad	wesen...				traditional_bed
34			adey	teze...	nlo...	abzia	medeb	tezeh...	zdele...	hisab	kikefil	ieye		traditional_bed
35			abeu	zne...	sebat	dima	nme...	tay	chigr	alewo	elom	hitoy	aqlelwo	traditional_bed
36	egrey	bzu...	may	tiha...	ab	liely	medeb	gdim	belku					traditional_bed
37	me...	slez...	em...	dima	kab	me...	med...	telael...	kreya...	fetena				traditional_bed
38	had...	ena...	kink...	ms...	am...	ab	med...	kewde...	kidgs...	wesen...				traditional_bed

arff screen shoots of the corpus 7 - 7 and 8 - 8 of words to the left side of target word and right side of the target word

File Edit View

1-1.arff * | 2-2.arff | 3-3.arff | 4-4.arff | 5-5.arff * | 6-6.arff | 7-7.arff *

Relation: book

No.	1: left7 String	2: left6 String	3: left5 String	4: left4 String	5: left3 String	6: left2 String	7: left1 String	8: target String	9: right1 String	10: right2 String	11: right3 String	12: right4 String	13: right5 String	14: right6 String	15: right7 String	16: class Nominal
35			ety	teje...	zelo	had...	sra...	med...	zala	atenak...	kiktslilu	ygbae				program
36				nEgri	israel	kab...	nab...	zmed...	mdri	melise	ayegls...	iye				program
37	ke...	elu	mel...	kulu	kem	etkEl	nety	zmed...	kea	zkilkleka	kem	zeyelbo	efelt	aleku		program
38							wer...	nme...	gziyat	gebere	tshay	nmEra...	tfelit			program
39	ke...	elu	me...	bhaqi	ke...	zha...	kike...	zmed...	ewin	kiqew...	iyu					program
40				ab	liEly	kula	mdriy	zmed...	mkri	eziy	iyu					program
41	key...	kea	qed...	neE...	nebyi	kitk...	dima	med...	bele							program
42		aba...	hiw...	zko...	neg...	nm...	qid...	medeb	kihly...	alewo						program
43						wei...	am...	medeb	tezey...	zteser...	ena	zeytes...	qeltifu	ayrdEon		program
44	nay	ethi...	gez...	wu...	nay	ha...	am...	medeb	awiju							program
45	am...	kilil	tigray	ato	abay	wel...	tmali	medeb	qetsali	hamu...	amet	mebra...	hibom			program
46	nay	eyfe...	me...	nay	qet...	afet...	srhu	medeb	merha	gibru	btewe...	hizbi	mkrbet	atsdiqu		program
47		hade	neg...	trfu	ena	kis...	zha...	medeb	kihlyo	ygbaE						program
48					nayta	ket...	ken...	medeb	qetsali	amet	aqribu					program
49					tem...	ab	ake...	medeb	tmhrty	kihlyom	tezarib...					program
50					nayta	hag...	mer...	medeb	Hizom	nhzbom	azarib...					program
51					we...	nEl...	me...	medeb	kihlyo	gdin	iyu					program
52					ab	srah	key...	medeb	tezey...	tserho	ytefEka					program
53							wer...	nmi...	gziyat	gebere	tshay	nmEra...	tiflit			program
54	mbr...	mE...	ent...	mE...	deb...	dima	kem	medeb	tetahi...	zeytefe...	zobatat	iyu				program
55					teze...	etiy	zha...	medeb	teqe...	keyhal...	kiterif	iyu				program
56	me...	gez...	nab	kor...	ztek...	kilte	kifly	medeb	zneb...	mede...	ewin	nab	hade	Aratn	kilte	traditional_bed
57			btht...	dass	zm...	seq...	me...	medeb	zelewo	geza	enthluw	nay	gubaE	geza	keme...	traditional_bed
58	nay	ges...	me...	kab	nay	aba...	me...	medeb	qereba	koynu	kab	enche...	zteser...	geza	iyu	traditional_bed
59							ab	medeb	mim...	abey	kemzd...	ksaE	ztefeniy	nguho	12	traditional_bed
60	bse...	akt...	ent...	aba	me...	alEl...	zdqi...	medeb	msAr...	nabty	meErefy	gasha	aketilo...	wesed...	abeu	traditional_bed
61	bm...	tetew	elom	tselot	ena...	ane...	ab	med...	koyne	teshef...	nzrey	dima	moyte	tselot	fthat	traditional_bed
62						buz...	me...	medeb	eka	enteh...	etiy	zeabeye	gn	doll	ybhal	traditional_bed
63							doll	medeb	qum...	2	metro	sfhatu	dima	bgmit	1	traditional_bed
64	wel...	kib...	me...	gob...	iyom	nm...	zke...	medeb	ngeza	dima	bqeliu	kab	ienche...	saerin	yserhu	traditional_bed
65	ab	zke...	adiy	ms...	nm...	nm...	zke...	medeb	ke	alo	diyu	elen	yzareba	nebera	btede...	traditional_bed
66								medeb	tezeyl...	dea	ab	mintay	tdqis	eka	belany	traditional_bed
67				ane	dima	nm...	zke...	medeb	enteli...	kihadr	iye	tezeyk...	gina	kikeyd	iye	traditional_bed
68	key...	ms2	aer...	nin...	gze	ab...	ena...	enaw...	kotas	buzuh	neger	enage...	nhadir	neberna	hiziy	traditional_bed
69				zha...	seb	ent...	gdin	medeb	trah	keykone	ab	arat	ewin	mdqas	ykeal	traditional_bed
70						nm...	zwuel	medeb	kekem	metenu	yfelale	iyu				traditional_bed
71		end...	tesf...	km...	kelo	ab	egri	med...	kof	elom	kulom	dekqom	ytsaw...			traditional_bed

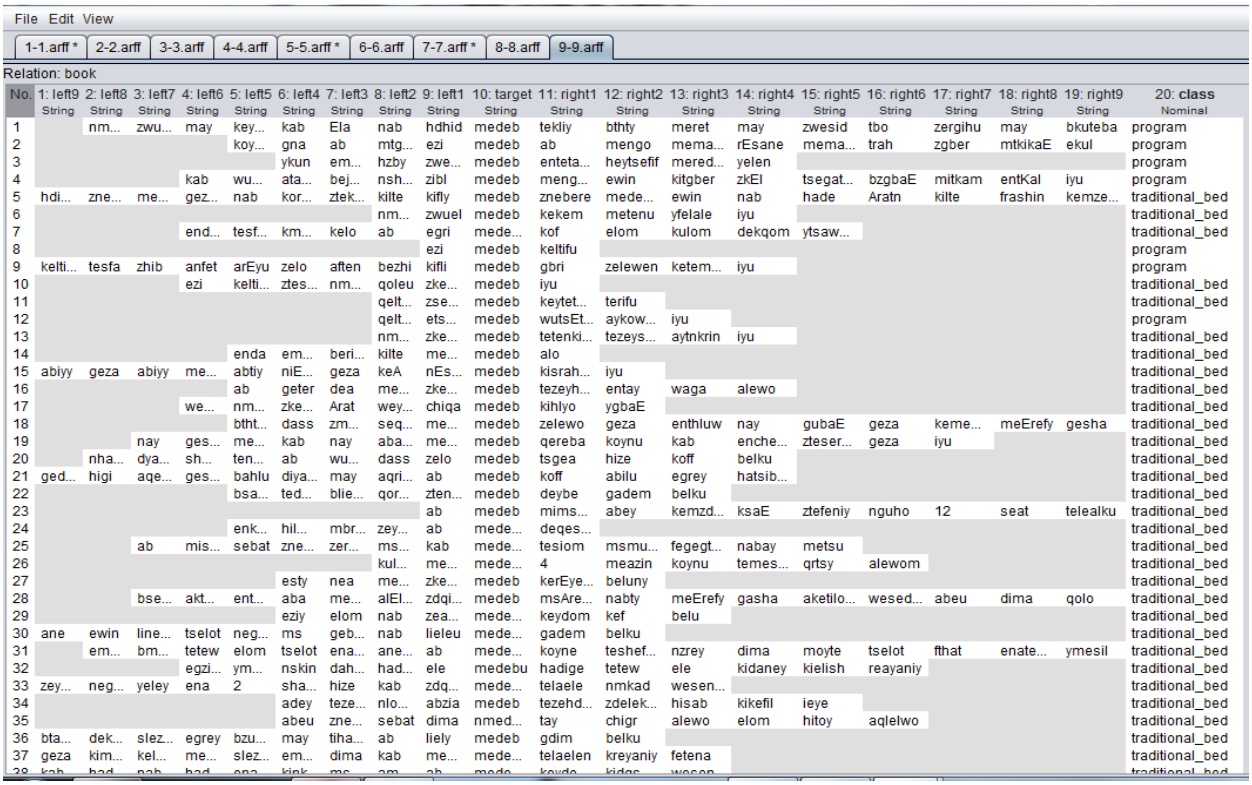
File Edit View

1-1.arff * | 2-2.arff | 3-3.arff | 4-4.arff | 5-5.arff * | 6-6.arff | 7-7.arff * | 8-8.arff

Relation: book

No.	1: left8 String	2: left7 String	3: left6 String	4: left5 String	5: left4 String	6: left3 String	7: left2 String	8: left1 String	9: target String	10: right1 String	11: right2 String	12: right3 String	13: right4 String	14: right5 String	15: right6 String	16: right7 String	17: right8 String	18: class Nominal
1	nm...	zwu...	may	key...	kab	Ela	nab	hdhid	medeb	tekliy	bthty	meret	may	zwesid	tbo	zergihu	may	program
2				koy...	gna	ab	mtg...	ezi	medeb	ab	mengo	mema...	rEsane	mema...	trah	zgber	mtkikaE	program
3					ykun	em...	hzby	zwe...	medeb	enteta...	heytsefif	mered...	yelen					program
4			kab	wu...	ata...	bej...	nsh...	zibl	medeb	meng...	ewin	kitgber	zkEl	tsegat...	bzgbaE	mitkam	entKal	program
5	zne...	me...	gez...	nab	kor...	ztek...	kilte	kifly	medeb	znebere	mede...	ewin	nab	hade	Aratn	kilte	frashin	traditional_bed
6							nm...	zwuel	medeb	kekem	metenu	yfelale	iyu					traditional_bed
7			end...	tesf...	km...	kelo	ab	egri	med...	kof	elom	kulom	dekqom	ytsaw...				traditional_bed
8								ezi	medeb	keltifu								program
9	tesfa	zhib	anfet	arEyu	zelo	aften	bezhi	kifli	medeb	gbri	zelewen	ketem...	iyu					program
10			ezi	kelti...	ztes...	nm...	qoleu	zke...	medeb	iyu								traditional_bed
11							qelt...	zse...	medeb	keytet...	terifu							traditional_bed
12							qelt...	ets...	medeb	wutsEt...	aykow...	iyu						program
13							nm...	zke...	medeb	tetenki...	tezeys...	aytnkrin	iyu					traditional_bed
14				enda	em...	beri...	kilte	me...	medeb	alo								traditional_bed
15	geza	abiyy	me...	abtiy	niE...	geza	keA	nEs...	medeb	kisrah...	iyu							traditional_bed
16				ab	geter	dea	me...	zke...	medeb	tezeyh...	entay	waga	alewo					traditional_bed
17			we...	nm...	zke...	Arat	wey...	chiqa	medeb	kihlyo	ygbaE							traditional_bed
18				btht...	dass	zm...	seq...	me...	medeb	zelewo	geza	enthluw	nay	gubaE	geza	keme...	meErefy	traditional_bed
19		nay	ges...	me...	kab	nay	aba...	me...	medeb	qereba	koynu	kab	enche...	zteser...	geza	iyu		traditional_bed
20	nha...	dya...	sh...	ten...	ab	wu...	dass	zelo	medeb	tsgea	hize	koff	belku					traditional_bed
21	higi	aqe...	ges...	bahlu	diya...	may	aqri...	ab	medeb	koff	abilu	egrey	hatsib...					traditional_bed
22				bsa...	ted...	blie...	qor...	zten...	medeb	deybe	gadem	belku						traditional_bed
23								ab	medeb	mims...	abey	kemzd...	ksaE	ztefeniy	nguho	12	seat	traditional_bed
24				enk...	hil...	mbr...	zey...	ab	med...	deqes...								traditional_bed
25		ab	mis...	sebat	zne...	zer...	ms...	kab	med...	tesiom	msmu...	fegegt...	nabay	metsu				traditional_bed
26							kul...	me...	med...	4	meazin	koynu	temes...	qrtsy	alewom			traditional_bed
27					esty	nea	me...	zke...	medeb	kerEye...	beluny							traditional_bed
28		bse...	akt...	ent...	aba	me...	alEl...	zdqi...	medeb	msAre...	nabty	meErefy	gasha	aketilo...	wesed...	abeu	dima	traditional_bed
29					eziy	elom	nab	zea...	med...	keydom	kef	belu						traditional_bed
30	ewin	line...	tselot	neg...	ms	geb...	nab	lieleu	med...	gadem	belku							traditional_bed
31	em...	bm...	tetew	elom	tselot	ena...	ane...	ab	med...	koyne	teshef...	nzrey	dima	moyte	tselot	fthat	enate...	traditional_bed
32			egzi...	ym...	nskin	dah...	had...	ele	med...	hadige	tetew	ele	kidaney	kielish	reayaniy			traditional_bed
33	neg...	yeley	ena	2	sha...	hize	kab	zdq...	med...	telaele	nmkad	wesen...						traditional_bed
34					adey	teze...	nlo...	abzia	medeb	tezehd...	zdelek...	hisab	kikefil	ieye				traditional_bed
35					abeu	zne...	sebat	dima	nme...	tay	chigr	alewo	elom	hitoy	aqlelwo			traditional_bed
36	dek...	slez...	egrey	bzu...	may	tiha...	ab	liely	medeb	gdim	belku							traditional_bed
37	kim...	kel...	me...	slez...	em...	dima	kab	me...	med...	telaelen	kreyaniy	fetena						traditional_bed

arff screen shoots of the corpus 9 – 9 of words to the left side of target word and right side of the target word



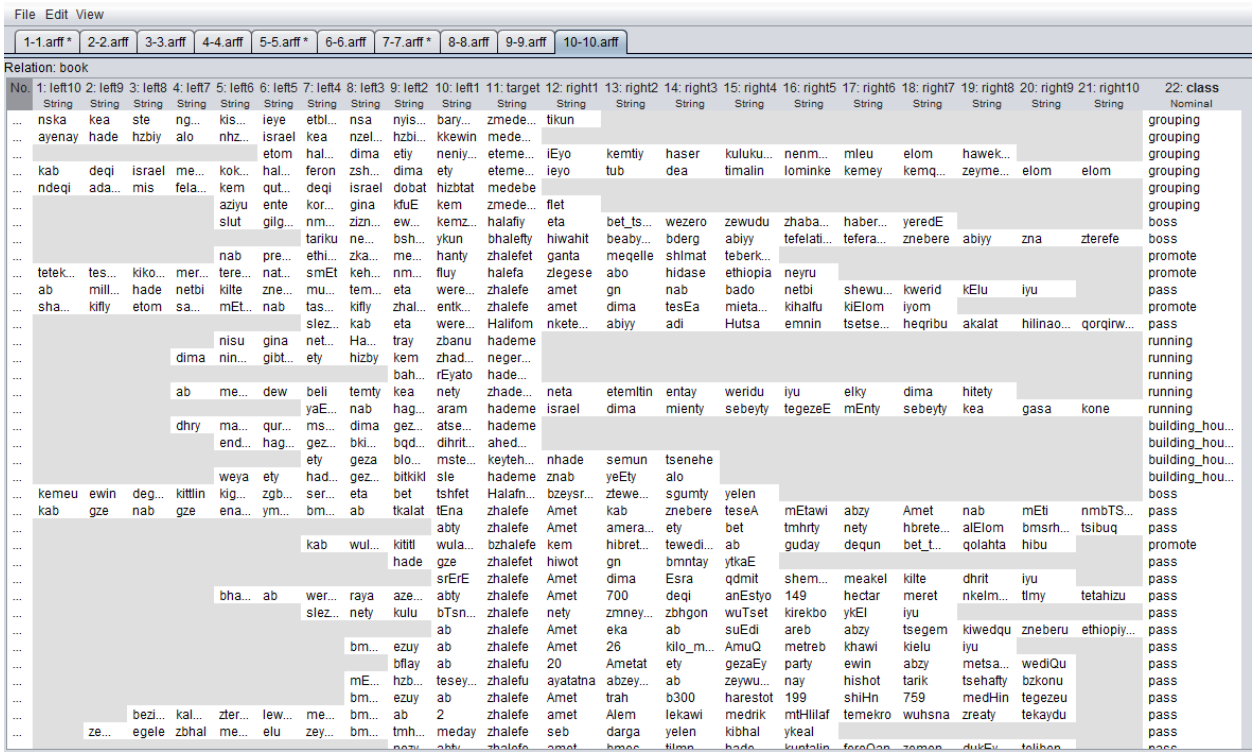
File Edit View

1-1.arff * | 2-2.arff | 3-3.arff | 4-4.arff | 5-5.arff * | 6-6.arff | 7-7.arff * | 8-8.arff | 9-9.arff

Relation: book

No.	1: left9	2: left8	3: left7	4: left6	5: left5	6: left4	7: left3	8: left2	9: left1	10: target	11: right1	12: right2	13: right3	14: right4	15: right5	16: right6	17: right7	18: right8	19: right9	20: class
	String	String	String	String	String	String	String	String	String	String	String	String	String	String	String	String	String	String	String	Nominal
1		nm...	zwu...	may	key...	kab	Ela	nab	hdhid	medeb	tekliy	bthty	meret	may	zwesid	tbo	zergihu	may	bkuteba	program
2					koy...	gna	ab	mtg...	ezi	medeb	ab	mengo	mema...	rEsane	mema...	trah	zgber	mtkikaE	ekul	program
3						ykun	em...	hzby	zwe...	medeb	enteta...	heytsefif	mered...	yelen						program
4				kab	wu...	ata...	bej...	nsh...	zibl	medeb	meng...	ewin	kitgber	zkEl	tsegat...	bzgbaE	mitkam	entKal	iyu	program
5	hdi...	zne...	me...	gez...	nab	kor...	ztek...	kilte	kifly	medeb	znebere	mede...	ewin	nab	hade	Aratn	kilte	frashin	kemze...	traditional_bed
6								nm...	zwuel	medeb	kekem	metenu	yfelale	iyu						traditional_bed
7				end...	tesf...	km...	kelo	ab	egri	mede...	kof	elom	kulom	dekqom	ytsaw...					traditional_bed
8									ezi	medeb	keltifu									program
9	kelti...	tesfa	zhib	anfet	arEyu	zelo	aften	bezhi	kifli	medeb	gbri	zelewen	ketem...	iyu						program
10				ezi	kelti...	ztes...	nm...	qoleu	zke...	medeb	iyu									traditional_bed
11								qelt...	zse...	medeb	keytet...	terifu								traditional_bed
12								qelt...	ets...	medeb	wutsEt...	aykow...	iyu							program
13								nm...	zke...	medeb	tetenki...	tezeys...	aytnkrin	iyu						traditional_bed
14					enda	em...	beri...	kilte	me...	medeb	alo									traditional_bed
15	abiyy	geza	abiyy	me...	abtiy	niE...	geza	keA	nEs...	medeb	kisrah...	iyu								traditional_bed
16					ab	geter	dea	me...	zke...	medeb	tezeyh...	entay	waga	alewo						traditional_bed
17				we...	nm...	zke...	Arat	wey...	chiqa	medeb	kihlyo	ygbaE								traditional_bed
18					btht...	dass	zm...	seq...	me...	medeb	zelewo	geza	enthluw	nay	gubaE	geza	keme...	meErefy	gesha	traditional_bed
19			nay	ges...	me...	kab	nay	aba...	me...	medeb	qereba	koynu	kab	enche...	zteser...	geza	iyu			traditional_bed
20		nha...	dya...	sh...	ten...	ab	wu...	dass	zelo	medeb	tsgea	hize	koff	belku						traditional_bed
21	ged...	higi	aqe...	ges...	bahlu	diya...	may	aqri...	ab	medeb	koff	abilu	egrey	hatsib...						traditional_bed
22					bsa...	ted...	blie...	qor...	zten...	medeb	deybe	gadem	belku							traditional_bed
23									ab	medeb	mims...	abey	kemzd...	ksaE	ztefeniy	nguho	12	seat	telealku	traditional_bed
24					enk...	hil...	mbr...	zey...	ab	mede...	deqes...									traditional_bed
25			ab	mis...	sebat	zne...	zer...	ms...	kab	mede...	tesiom	msmu...	fegegt...	nabay	metsu					traditional_bed
26								kul...	me...	mede...	4	meazin	koynu	temes...	qrtsy	alewom				traditional_bed
27						esty	nea	me...	zke...	medeb	kerEye...	beluny								traditional_bed
28			bse...	akt...	ent...	aba	me...	alEl...	zdqi...	medeb	msAre...	nabty	meErefy	gasha	aketilo...	wesed...	abeu	dima	qolo	traditional_bed
29						eziy	elom	nab	zea...	mede...	keydom	kef	belu							traditional_bed
30	ane	ewin	line...	tselot	neg...	ms	geb...	nab	lieleu	mede...	gadem	belku								traditional_bed
31		em...	bm...	tetew	elom	tselot	ena...	ane...	ab	mede...	koyne	teshef...	nzrey	dima	moyte	tselot	fthat	enate...	ymesil	traditional_bed
32				egzi...	ym...	nskin	dah...	had...	ele	medebu	hadige	tetew	ele	kidaney	kielish	reayaniy				traditional_bed
33	zey...	neg...	yeley	ena	2	sha...	hize	kab	zdq...	mede...	telaele	nmkad	wesen...							traditional_bed
34					adey	teze...	nlo...	abzia	medeb	tezehd...	zdelek...	hisab	kikefil	ieye						traditional_bed
35					abeu	zne...	sebat	dima	nmed...	tay	chigr	alewo	elom	hitoy	aqlelwo					traditional_bed
36	bta...	dek...	slez...	egrey	bzu...	may	tiha...	ab	liely	medeb	gdim	belku								traditional_bed
37	geza	kim...	kel...	me...	slez...	em...	dima	kab	me...	mede...	telaelen	kreyaniy	fetena							traditional_bed
38	kab	had	nab	had	ena	kink	ms	am	ab	mede	kayde	kidge	weaen							traditional_bed

arff screen shoots of the corpus 10 - 10 of words to the left side of target word and right side of the target word



File Edit View

1-1.arff * | 2-2.arff | 3-3.arff | 4-4.arff | 5-5.arff * | 6-6.arff | 7-7.arff * | 8-8.arff | 9-9.arff | 10-10.arff

Relation: book

No.	1: left10	2: left9	3: left8	4: left7	5: left6	6: left5	7: left4	8: left3	9: left2	10: left1	11: target	12: right1	13: right2	14: right3	15: right4	16: right5	17: right6	18: right7	19: right8	20: right9	21: right10	22: class
	String	String	String	String	String	String	String	String	String	String	String	String	String	String	String	String	String	String	String	String	String	Nominal
...	nska	kea	ste	ng...	kis...	ieye	etbl...	nsa	nyis...	bary...	zmede...	tikun										grouping
...	ayenay	hade	hzbiy	alo	nhz...	israel	kea	nzel...	hzbi...	kkewin	mede...											grouping
...						etom	hal...	dima	etiy	neniy...	eteme...	iEyo	kemtiy	haser	kuluku...	nenm...	mleu	elom	hawek...			grouping
...	kab	deqi	israel	me...	kok...	hal...	feron	zsh...	dima	ety	eteme...	ieyo	tub	dea	timalin	lominke	kemey	kemq...	zeyme...	elom	elom	grouping
...	ndeqi	ada...	mis	fela...	kem	qut...	deqi	israel	dobat	hizbtat	medebe											grouping
...					aziyu	ente	kor...	gina	kfuE	kem	zmede...	flet										grouping
...					slut	gilg...	nm...	zizn...	ew...	kemz...	halafiy	eta	bet_ts...	wezero	zewudu	zhaba...	haber...	yeredE				boss
...							tariku	ne...	bsh...	ykun	bhalefty	hiwahit	beaby...	bderg	abiyy	tefelati...	tefera...	znebere	abiyy	zna	zterefe	boss
...					nab	pre...	ethi...	zka...	me...	hanty	zhalefet	ganta	meqelle	shlmat	teberk...							promote
...	tetek...	tes...	kiko...	mer...	tere...	nat...	smEt	keh...	nm...	fluy	halefa	zlegese	abo	hidase	ethiopia	neyru						promote
...	ab	mill...	hade	netbi	kilte	zne...	mu...	tem...	eta	were...	zhalefe	amet	gn	nab	bado	netbi	shewu...	kwerid	kElu	iyu		pass
...	sha...	kifly	etom	sa...	mEt...	nab	tas...	kifly	zhal...	entk...	zhalefe	amet	dima	tesEa	mieta...	kihalfu	kiElom	iyom				promote
...						slez...	kab	eta	were...	Halifom	nkete...	abiyy	adi	Hutsa	emnin	tsetse...	heqribu	akalat	hilinao...	qorqirw...		pass
...					nisu	gina	net...	Ha...	tray	zbanu	hademe											running
...				dima	nin...	gibt...	ety	hizby	kem	zhad...	neger...											running
...									bah...	rEyato	hade...											running
...				ab	me...	dew	beli	temty	kea	nety	zhade...	neta	etemltin	entay	weridu	iyu	elky	dima	hitety			running
...							yaE...	nab	hag...	aram	hademe	israel	dima	mienty	sebeyty	tegezeE	mEnty	sebeyty	kea	gasa	kone	running
...				dhry	ma...	qur...	ms...	dima	gez...	atse...	hademe											building_hou...
...					end...	hag...	gez...	bki...	bqd...	dihrit...	ahed...											building_hou...
...							ety	geza	blo...	mste...	keyteh...	nhade	semun	tsenehe								building_hou...
...					weya	ety	had...	gez...	bitkikl	sle	hademe	znab	yeEty	alo								building_hou...
...	kemeu	ewin	deg...	kittlin	kig...	zgb...	ser...	eta	bet	tshfet	Halafn...	bzeysr...	ztewe...	sgumty	yelen							boss
...	kab	gze	nab	gze	ena...	ym...	bm...	ab	tkalat	tEna	zhalefe	Amet	kab	znebere	teseA	mEtawi	abzy	Amet	nab	mEti	nmbTS...	pass
...										abty	zhalefe	Amet	amera...	ety	bet	tmhrty	nety	hbrete...	alElom	bmsrh...	tsibuq	pass
...							kab	wul...	kititl	wula...	bzhalefe	kem	hibret...	tewedi...	ab	guday	dequn	bet_t...	qolahta	hibu		promote
...									hade	gze	zhalefet	hiwot	gn	bmntay	ytkaE							pass
...										srErE	zhalefe	Amet	dima	Esra	qdmit	shem...	meakel	kilte	dhrit	iyu		pass
...					bha...	ab	wer...	raya	aze...	abty	zhalefe	Amet	700	deqi	anEstyo	149	hectar	meret	nkelm...	tlmy	tetahizu	pass
...							slez...	nety	kulu	bTsn...	zhalefe	nety	zmney...	zbhgon	wuTset	kirekbo	ykEl	iyu				pass
...									ab	zhalefe	Amet	eka	ab	suEdi	areb	abzy	tsegem	kiwedqu	zneberu	ethiopiy...	pass	
...								bm...	ezuy	ab	zhalefe	Amet	26	kilo_m...	AmuQ	metreb	khawi	kielu	iyu			pass
...									bflay	ab	zhalefu	20	Ametat	ety	gezaEy	party	ewin	abzy	metsa...	wediQu		pass
...								mE...	hzb...	tesey...	zhalefu	ayatatna	abzey...	ab	zeywu...	nay	hishot	tarik	tsehafty	bzkonu		pass
...								bm...	ezuy	ab	zhalefe	Amet	trah	b300	harestot	199	shiHn	759	medHin	tegezeu		pass
...			bezi...	kal...	zter...	lew...	me...	bm...	ab	2	zhalefe	amet	Alem	lekawi	medrik	mtHlilaf	temekro	wuhsna	zreaty	tekaydu		pass
...		ze...	egele	zbhal	me...	elu	zey...	bm...	tmh...	meday	zhalefe	seb	darga	yelen	kibhal	ykeal						pass
...									nezy	abty	zhalefe	amet	bmes	tilmn	hade	kuntalin	foreQan	zemen	dukEy	teliben		pass

Section 3

Appendex 13 Sample list of Tigrigna sentence corpus examples

1) ንመስኖ ዝውዕል ማይ ከይባኽን ካብ ዒላ ናብ ሕድሕድ **መደብ** ተኽሊ ብትሕቲ መሬት ማይ ዝወስድ ትቦ ዘርጊሑ ማይ በቐጠባ ይጥቀም።
2) ኾይኑ ግና ኣብ ምትግባር እዙ **መደብ** ኣብ መንነ መማህራንን ርእሳነ መማህራንን ጥራሕ ዝግበር ምትኽኻእ እኹል ኣይኮነን።
3) ይኹን እምበር ህዝቢ ዝወነኖ **መደብ** እንተተታሒዙ ዘይፀፍፍ መረዳእታ የለን።
4) ካብ ውሽጣዊ ኣታዊና በጀትና ንሸፍን ዝብል **መደብ** መንግስቲ እውን ክትግበር ዝኽእል ፀጋታትካ ብዝግባእ ምጥቃም እንትከኣል እዩ።
5) እዚ **መደብ** ቀልጢፉ ተስፋ ዜህብ ኣንፊት ኣርእዩ ዘሎ ኣብተን በዝሒ ከፋሊ ግብሪ ዘለወን ከተማታት እዩ።
6) እዚኣም ድማ ነቲ ሕዚ ኣብ ኢትዮጵያ ዘሎ **መደብ** ምግቢ ንስራሕ መሰረት ዝኾንዎ እዮም።
7) ቁልፊ **መደብ** እውን ዓቕሚ ሰራሕተኛን ኣመራርሓን ብምዕባይ ልምዓታዊ ሰራዊት ምህናፅ እዩ።
8) ነዚ ምስኻዕ ምእንታን ክከኣል ብዩኒቨርስቲ መቐለ ኮሌጅ ጥዕና እንስሳ ሰፊሕ **መደብ** ተታሒዘ ክስረሐሉ ጀሚሩ ኣሎ።
9) 48 ክፍልታት ዝሓዘ ሽዱሽተ ብሎኳት ምምሕዳር ከተማ መቐለን ቢሮ ቴሞትስን **ብዜመደብዎ** 12 ነጥቢ ሽዱሽተ ሚልዮን ብር ተሃኒፁ ኣብቲ እዋን ተመሪቑ እዩ።
10) ምኽንያቱ ብቐዕ ትልሚ ምድላው ልዕሊ 50 ሚኢታዊ **መደብ** ምዕዋት ስለ ዜኾነ።
11) መቐለ ሓምለ 16/2007(ድወት) ኣብዚ ዓመት ኣብ በሄራዊ ሊግ ምደባ ሰሜን ሻምፒን ብምኽን ዓበይቲ ናብፕሪመር ሊግ ኢትዮጵያ ዝኻየድ መፃራይ **ዝሓለፈት** ጋንታ መቐለ ሽልማት ተበርኪቱላ።
12) ማሕበራውን ፖለቲካውን ዘፈራት ልምዓት ተጠቀምትን ተሳተፍትን ክኾኑን መሪሕነት ተረኪቦም ናተይነት ስምዒት ከሕድሩን ንመናእሰይ ፉሉይ **ሓለፋ** ዝለገሰ ኣቦ ሕዳሰ ኢትዮጵያ ነይሩ።
13) ኣብዙ ዝተርኣየ ለዉጢ ዜምልከት ቅድሚ ትልሚ ዕቤትን ስግግርን ኣብ 2001 ዓ/ም ሓደ ነጥቢ 27 ዜነበረ ምቁራፅ ተምሃሮ እታ ወረዳ ኣብ **ዝሓለፈ** 2006 ዓ/ም ግን ናብ ባዶ ነጥቢ 75 ክወርድ ክኢሉ እዩ።
14) ከም ኣብነት ኣብ 2003 ዓ/ም ብሄራዊ ፈተና ካብ ዜወሰዱ ተምሃሮ ሻሙናይ ክፍሊ እቶም 81 ምኢታዊ ናብ ታሽዓይ ክፍሊ ዜሓለፉ እንትኾኑ፤ ኣብ **ዝሓለፈ** 2006 ዓ/ም ድማ 96 ምኢታዊ ክሓልፉ ክኢሎም እዮም።
15) ትካላት ጥዕና ንዜግበር ወሊድ ኣብ **ዝሓለፈ** ዓመት ካብ ዜነበረ 97 ምኢታዊ ኣብዙ ዓመት ናብ ምኢቲ ምኢታዊ ንምብፃሕ ምሉእ ምድላው ከም ዝተገበረ ድማ ንምፍላጥ ተኻኢሉ ኣሎ።
16) ኣብቲ **ዝሓለፈ** ዓመት ኣመራርሓ እቲ ቤት ትምህርቲ ነቲ ሐ/ሰብ ኣለዓዒሎም ብምስርሖምን ፅቡቕ ለውጢ ብምምዘጋቦምን ር/መምህርን ም/ር/መምህርን ብብርኪ ወረዳ ተሸላምቲ ክኾኑ ኣኽኢልዎም እዩ።

17) ካብ ውልቀ ክትትል ውላዱ **ብዝሓለፈ** ከም ሕብረተሰብ ተወዲቡ ኣብ ጉዳይ ደቁን ቤት ትምህርቱን ቆላሕታ ሂቡ ምስታፉ ካልኦት እውን ከወስድዎ ዜግባእ ተመኩሮ እዩ።።

18) **ዝሓለፈት** ህይወት ግን ብምንታይ ይትካእ?

19) ስርርዕ **ዝሓለፈ** ዓመት ድማ 20 ቅድሚት፣ ሽሞንተ ማእኸል ክልተ ድሕሪት እዩ።።

20) ብሓፈሻ ኣብ ወረዳ ራያ ዓዘቦ ኣብቲ **ዝሓለፈ** ዓመት 733 ደቂ ኣንስትዮ 149 ሄክታር መሬት ንኽልምዓ ትልሚ ተታሒዘ

21) ንሱ ግና ነጸላኡ ሐዲጉ ጥራይ ዝባኑ **ሃደመ**።

22) ድማ ንንጉስ ግብጺ እቲ ህዝቢ ኸም **ዝሃደመ** ነገርዎ።

23) ባሕሪ ርእያቶ **ሃደመት**።

24) ካብ ክብረትኪ ውረዲ፡ ኣብ ኣጻምእ ተቐመጢ፡ ኣቲ ኣብ ኣሮኤር እትነብሪ ዘሎኺ፡ ኣብ መገዲ ደው በሊ ጠምቲ ኸኣ፡ ነቲ **ዝሃደመን** ነታ እተምልጥን፡ እንታይ ወሪዱ እዩ ኢልኪ ድማ ሕተቲ።

25) ያእቆብ ናብ ሃገር ኣራም **ሃደመ**፣ እስራኤል ድማ ምእንቲ ሰበይቲ ተገዝኤ፣ ምእንቲ ሰበይቲ ኸኣ ጓሳ ኼነ።

26) ሽዑ እቶም ሰባት ንሱ ካብ ገጽ እግዚኣብሄር ርሒቑ ኸም **ዝሃደመ**፣ ባዕሉ ስለ ዝነገሮም፣ ፈለጡ፣ ዓብዩ ፍርሃትውን ፈርሁ እሞ፣ ስለምንታይ ደኣ እዚ ገበርካ ከኣ በልዎ።

27) ሙሴ ድማ ብዛዕባ እዚ ዘረባ እዚ **ሃደመ**፡ ኣብ ምድሪ ምድያን ድማ ስደተኛ ኼነ፡ ኣብኡ ኸኣ ክልተ ኣወዳት ወለደ።

28) እታ ሰበይቲ ድማ፡ ሽሕን ክልተ ሚእትን ስሳን መዓልቲ ኺምግብዋ፡ ናብቲ ኣምላኽ ዘዳለወላ ቦታ ናብ በረኻ **ሃደመት**።

29) ሽዑ ሳራይ ኣዋረደታ፡ ንሳ ኸኣ ካብ ቅድሚኣ **ሃደመት**።

30) ዘለዎ ዅሉ ሒዙ ኸኣ **ሃደመ**፡ ተንሲኡ ድማ ርባ ተሳገረ፡ ገጹ ኸኣ ናብ ከረን ጊልዓድ ኣቢሉ ኣቕንዔ።

31) ኣብ ሳልሰይቲ መዓልቲ ኸኣ ያእቆብ ከም **ዝሃደመ** ንላባን ነገርዎ።

32) ንሳ ድማ ክዳኑ ኣብ ኢዳ ኸም ዝሐደገን ንግዳም ከኣ ከም **ዝሃደመን** ምስ ረኣየት፡ ንስድራ ቤታ ጸዊዓ፡ ርእዩ፡ ኪሰሓቐልናስ እብራዊ ሰብኣይ ኣእትዩልና፡ ንሱ ምሳይ ኪድቅስ ናባይ ኣተወ፡ ኣነ ኸኣ ዓው ኢለ ኣእዌኹ።

33) ኰነ ድማ፡ ቃለይ ዓው ኣቢለ ምስ ኣእዌኹ፡ ክዳኑ ሐዲጉለይ ንግዳም **ሃደመ**።

34) ሙሴ ግና ካብ ቅድሚ ፈርኦን **ሃደመ**፡ ኣብ ምድሪ ሚድያን ድማ ተቐመጠ፡ ኣብ ጥቓ ዔላ ኸኣ ኮፍ በለ።

35) ናብ ምድሪ ደርበያ፡ ተመን ከኣ ኰነት። ሙሴ ድማ ካብ ቅድሚኡ **ሃደመ**።

36) እቲ ቐታሊ ኻብ ዶብ እታ **ዝሃደመላ** ኸተማ መዕቈቢቱ እንተ ወጸ ግና፡እቲ ፈዳይ ደም ከኣ ኣብ ወጻኢ ዶብ እታ ኸተማ መዕቈቢቱ እንተ ረኸቦ፡ እቲ ፈዳይ ደም ድማ ነቲ ቐታሊ እንተ ቐተሎ፡ ዕዳ ደም የብሉን።

37) ብ ውሽጢ ዓዲ ድማ ኣብ **ዝሓለፈ** ሕታምና ዝረኣናዮ ኣብ ሰፋሕቲ በረኻታት ብሄረሰብ ኮንሶ ዜተኻየደ ልውውጥ ተመከሮ ምውሳድ ይከኣል።

38) ኣብ **ዝሓለፈ** ሕታም ጋዜጣና ምስ ተጋዳላይ ከያኒ ጠርጣራው ስቡሕ ተላሊና ፤ ጠርጣራው ዝብል ሽም ኣብ ገድሊ ከም ዝወፅአሉን ናይዙ ኣሰያይማን፣ ጠርጣራውን ኪነትን ከመይ ከም ዝተራኸቡ ኣብ ዝብሉን ካልኦትን ዚዕበታት ብቐዳማይ ክፋል ኣተኣናጊድና ነይርና።

39) ብፍላይ ኣብቲ **ዝሓለፈ** ዕጥቃዊ ቓልሲ ሓንቲ ደርፊ ኣብ ስርዓት ደርግ እተብፅሑ ፖለቲካዊ ኪሳራ ፍሉጥ እዩ ነይሩ።

40) እቲ **ዝሓለፈ** ክፋል እንታይነት ክፍሊ ስርዒት፣ ቀንዲ ዕማማቱ፣ ኣወዳድብኡ፣ ከይዲ ምልመላ ኣባላቱ፣ ኣብ እዋን ቀይሕ ራዕዲ ኣንፀላልዩዎ ዝነበረ ሓደጋ ወዘተ ቀሪቡ ነይሩ።

41) እዙ ብዓይኒ ኣፈፃፅማ **ዝሓለፈ** ዓመት እንትረአ ሕብረተሰብ ንፅሬት እታ ከተማ ዘለዎ ግንዚበ መርኣያ ዝኾነ ተግባር ከም ዝፈፀመ ይውሰድ።

42) ሕሉፋትስ ሓንሳብ ስለ**ዝሓለፉ** ክልተ ግዘ ዘዐዝብ ዓይኒ ኣይረኽቡን።

43) በዚ ከይዲ **ዝሓለፈ** ጉዳይ ልክዕ ኣብ ስሩዕ ቤት ፍርዲ ከም ዝሓለፈ ውሳነ መግደዳይ ፀባይ ኣለዎ።

44) ኣብራሃም ድማ በታ ምድሪ ኽሳዕ ቦታ ሴኬም ክሳዕ ድዋታት ሞሬ **ሓለፈ**።

45) ካብቲ ናይ መደረብታ ዓለባታት ድንኳን ዝተረፈ ሕልፊ ፈርቓ እቲ **ዝሓለፈ** ዓለባ ብድሕርቲ ማሕደር ይጀርበብ።

46) እቲ ኻብ ምንዋሕ ናይቲ መደረብታ ዓለባታት ድንኳን **ዝሓለፈ**፡ ምእንቲ ኪኸድኖ፡ ብኽልተ ሸነኹ፡ እመት በዚ እመት በቲ፡ ነቲ ማሕደር በዝን በትን ይጀርበብ።

47) ብዘሎ እቲ ህዝቢ ፈጺሙ ንዮርዳኖስ ክሳዕ ዚሳገሮ፡ ኵሉ እስራኤል በቲ ንቐጽ **ሓለፈ**።

48) ሸዑ ጴድቅያስ ወዲ ከናዓና ቐሪቡ ንሚክያስ ኣብ ምዕጉርቱ ጸፍዖ እሞ፣ እቲ መንፈስ እግዚኣብሄርሲ ንዓኻ ኺዛረብ በየናይ መገዲ እዩ ኻባይ **ዝሓለፈ** በሎ።

49) ኢሰፓ እውን ከምዘይመፅእ ንፈልጥ፣ ከም መጠን መርገፂ ግና ብዕቱብ ዝኣመንናሉ ንኽመፁ እውን መልእኽቲ **ሓሊፉ** እዩ።።

50) ይኹን እምበር ውድብና ህወሓት ነዘ ኩለ ፀበባን ከባ ፀሊእትን ብትዒግስትን ውሕላሌ ተፃዋርነትን እና**ሓለፈ** ናብ ዝበለፀ ታሪክ ዛተሰጋገረ ውድብ እዩ።።

51)

52) እታ ኣሚና "መሕደሪ ኣለኒ" ኢላ ናብቲ ገዛ ዝኣተወት ህይወት ህይወታ ክተድሕን ዘካየደቶ ቃልሲ ከይተዓወተ ተሪፋ ብኢድ ፍቐረኛኣ **ሓለፈት**።።

53) ዓብዪ ሓው ድማ ኩሉ ጊዘ ዝበለፀ ሓላፊነት ኣለዎ።።

54) ስለዚ ዝበለፀ **ሓላፍነት** ካብ መረብ ንደቡብ ኣብ ዘለና ተጋሩ ዝወደቀ እዩ።።

55) ካብኡ **ሓሊፉ** እውን ብፍቓድ ኡቶማን ቱርኪ ክፋል ኢራን፣ ፓኪስታን ይገዝኡ ነይሮም።።

56) ….እንኮላይ እቶም ብደሞም ናይቶም **ሕሉፋት** ስርወ-መንግስታት ኣባላት ዝኾኑ..

57) መራሕቱ ብፍላይ ኣብ ዝሓለፉ 4 ሚእቲ ዓመታት ብፍላጥን ብዘይፍላጥን ብተደጋጋሚ ብዝፈፀሙዎ ስትራቴጂያዊ ጌጋታት ህዝቢ ኩርድ ብዙሕ ዋጋ ይኽፍል ኣሎ

58) ሃገር ከምልሱ ንዝሓለፉ 90 ዓመታት ኣብ ቃልሲ ኣለዉ

59) ኣብ **ዝሓለፉ** 120 ዓመታት ግን ውሽጣዊ ሕመቐ ዝተበለፀ ግዳማዊ ሓይልታት

60) እቲ ሎሚ ኣብ 30ታት፣ 40ታት ዘሎ ዕድሚኡ ንቐምነገር፣ **ንሓላፍነት**፣ ንሃፍቲ፣ ንስልጣን ወ.ዘ.ተ ዝበቐዐ ትግራዋይ ዳርጋ ኣብ ኩሉ የለን